I/O 別冊

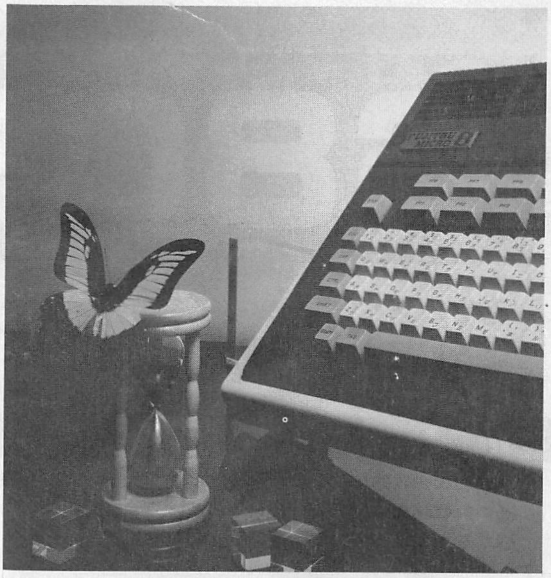
FM-8活用研究



I/O 別冊

FM-8活用研究

CONTENTS



システム／アプリケーション編

ゲーム編

ハード編

資料編

6809の特徴

6809の一般的な特徴は，すでに本誌を始めとする各種マイコン雑誌上に発表されているので，ここでは，6800と比較しながら述べていきます．

6809の内部レジスタ

6809の内部レジスタの様子を図1に示します．8ビット長のレジスタとしては，

❶アキュムレータ（accumulator）　A
❷アキュムレータ（accumulator）　B
❸ダイレクトページ・レジスタ　DPR
(direct page register)
❹コンディションコード・レジスタ　CCR
(condition code register)

以上の4本があり，特にAとBは，Aを上位バイトとして連結し，**アキュムレータD**と呼んで，16ビット長のレジスタとしても使用可能です．

また，16ビット長のレジスタとして，

❶インデックス・レジスタ　X
(index register)
❷インデックス・レジスタ　Y
❸スタック・ポインタ　S
(stack pointer)
❹スタック・ポインタ　U
❺プログラム・カウンタ　PC
(program counter)

以上の5本があります．これら各レジスタの機能について，以下説明を加えます．

(1)　アキュムレータ(A, B, D)

6800と同様A，B2本のアキュムレータを持っていますが，Aを上位バイト，Bを下位バイトとして連結し，**Dレジスタ**と呼んで16ビットの演算が可能となっています．

また，8ビット×8ビットの符号なし**乗算**が実行可能であることも注目すべき点です．

(2)　ダイレクトページ・レジスタ(DPR)

6800において，ダイレクト・アドレッシングが使用可能な領域は0000～00FFの256バイトだけでしたが，6809では，**ダイレクトページ・レジスタ**(以下DPR)を設けることで，64Kバイトすべてのアドレス空間をダイレクト・アドレッシングできるようになりました．

DPRは8ビットのレジスタで，ダイレクト・アドレッシング命令を実行したときに，A_8～A_{15}にDPRの内容が出力されます．

CPUがリセットされたときには，6800と同等な働きをするようにDPRの内容はクリアされますが，プログラム中で任意の値をセットすることで，ダイレクト・ページを任意の番地に設定することが可能です．

(3)　インデックス・レジスタ(X, Y)

6800ではインデックス・レジスタはX1本だけでしたが，Yが追加されるとともにインデックス修飾の機能が大幅に拡張されました．

詳しくはソフトウェアの項で述べますが，スタック・ポインタS，Uにもインデックス・レジスタと同様な働きをさせることも可能です．

(4)　スタック・ポインタ(U, S)

スタック・ポインタS(以下SP)は6800と同様に，サブルーチン・コールやインタラプト処理のときに自動的に使用されるハードウェア・スタックのポインタです．

スタック・ポインタU(以下UP)は，このような仕事からは解放されているユーザー・スタック・ポインタです．サブルーチンへの引数引き渡しなどに有効に使用可能となります．

また，6809のすべての内部レジスタ(CCR, A, B, DPR, X, Y, U, S, PC)は，SPおよびUPの示すスタックへ退避可能となりました．

6809の命令実行に要するマシン・サイクル数は，そのほとんどが6800と比較し同等か，1サイクル程度遅くなっていますが，2本のインデックスと2本のスタック・ポインタを有効に使用すれば，マシン・サイクルの遅さを補って余りあるものと期待されます．

(5)　プログラム・カウンタ(PC)

6800と同様16ビットのレジスタですが，ある種の命令では，PCをインデックス・レジスタと同様な使い方が可能です．

この機能によって，記憶場所のアドレスに依存しないプログラム――**ポジション・インディペンデント**(Position Independent)なプログラム――を作ることもできます．

図1　6809の内部レジスタ

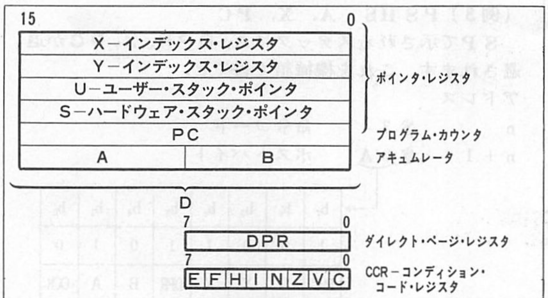
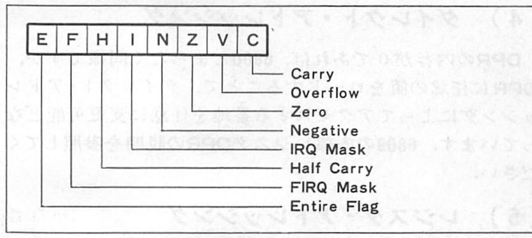


図2　CCRのフォーマット



（6）　コンディションコード・レジスタ（CCR）

図2のように，6800では使用されていなかった上位2ビットが以下のような働きをしています．

ア．FIRQ Mask（ビット6）Fフラグ

FIRQとはファースト インタラプト・リクエスト(fast interrupt request)のことです．

6800のIRQは，すべての内部レジスタ（CCR, B, A, X, PC）をスタックへ退避するために，処理時間が長くかかるという欠点がありました．これに対し，6809ではFIRQという端子を設け，この端子からのインタラプトには，CCRとPCのみをスタックへ退避します．Fフラグを“1”とすることによって，FIRQを拒否することもできます．

また，Fフラグは，NMI, FIRQ, SWIおよびRESETにより“1”にセットされます．

イ．Entire Flag（ビット7）Eフラグ

前項で述べたように，6809においては，インタラプト処理の際，内部レジスタのスタックへの退避方法に2種類あります．このため，インタラプト処理からリターンする際に，2種あった中のどちらであるかを区別しなければなりません．

Eフラグはこのために使用されるもので，“1”ならば，すべてのレジスタをスタックから取り出してリターンします．逆に“0”ならば，CCRとPCの値のみスタックから取り出してリターンすることになります．

6809の命令セット

6809の実行可能な命令セットを表に示します．機械語レベルでは，一部6800と異なっており，特に，インデックス・レジスタおよびスタック・ポインタ操作に関する命令は，ニモニックが同じでも，機械語コードは違うので注意が必要です．

基本命令数は6800の72から59種類へと減少したものの，アドレッシング・モードの変化による組み合わせ数では197から，1,464種類となり，これにともなって，1つの命令に必要なバイト数は1～4バイトまでと種類が増えました．

命令実行に必要なマシン・サイクル数では，ダイレクトおよびエクステンド・アドレッシング・モードで，残念ながら6800に比べ1サイクル余分に必要となっています．

また，インデックス・アドレッシングおよびレジスタ・アドレッシングの各モードでは，命令コードに引き続く次のバイトを**ポストバイト**（POST BYTE）と呼び，アドレッシングに使用されるレジスタやデータが各種各様に変化し，これによって，1命令に必要なバイト数や，命令実行に必要なマシン・サイクル数が変化します．

6809のアドレッシング・モード

6809で使用可能なアドレッシング・モードには，次のようなものがあります．

❶インヘレント(inherent)
❷イミディエート(immediate)
❸エクステンド(extended)
（およびエクステンディット・インダイレクト）
❹ダイレクト(direct)
❺レジスタ(register)
❻インデックス(indexed)
ア．ゼロ・オフセット(zero offset)
イ．コンスタント・オフセット(constant offset)
ウ．アキュムレータ・オフセット
(accumulator offset)
エ．オートインクリメント/デクリメント
(auto increment/decrement)
オ．インデックスド・インダイレクト
(indexed indirect)
❼リラティブ(relative)
ア．ショート/ロングブランチ
(short/long branching)
イ．プログラムカウンタ・リラティブ
(program counter relative addressing)

以上のうち，❶と❷は6800と同様なので説明は省略します．

（3）　エクステンディッド・アドレッシング

6800にはない機能であるエクステンディッド・インダイレクトの例を示します．

〔例1〕　エクステンデッド・インダイレクト

アドレス(ラベル)	コード	
××××	LDA	[CAT]
⋮	⋮	
CAT	10	
CAT+1	52	
⋮	⋮	
1052	41	

上の例で，[　]の記号は，6809用アセンブラでインダイレクト・アドレッシングを表わします．[　]がなければ，AレジスタにはCAT番地の内容である10が入りますが，[　]が付くと，CATおよびCAT＋1番地の内容が新たなアドレスになります．この結果Aレジスタには，1052番地の内容である41がロードされることになります．

(4)　ダイレクト・アドレッシング

DPRの内容が0であれば，6800とまったく同様ですが，DPRに任意の値をロードすることで，ダイレクト・アドレッシングによってアクセスする番地を任意に変更可能となっています．6809の内部レジスタDPRの説明を参照してください．

(5)　レジスタ・アドレッシング

6800における，内部レジスタ間のデータ転送はTAB，TBA，TSX，TXSの4個のみでしたが，6809では，ビットサイズの等しいレジスタ同士であれば，データの転送および交換が可能です．

転送はTFR($1F, transfer)，交換はEXG($1E, exchange)命令で行ない，これらの命令に続く次のバイト(先ほど説明したポストバイト)が，転送または交換に使用されるレジスタを示します．ポストバイトの上位4ビットと下位4ビットで，それぞれ1個ずつのレジスタを表わしますが，このときのビット・パターンと，レジスタの関係を図3に示します．

図　3　TFR, EXG命令のポストバイト

0000－D	0101－PC
0001－X	1000－A
0010－Y	1001－B
0011－U	1010－CCR
0100－S	1011－DPR

〔例2〕
TFR　A, DPR
はAの内容をDPRに転送を意味し，これを機械語で表わすと，

アドレス	$1F	
n	$1F	命令コード
n＋	$8B	ポストバイト

(8 → DPR，B → A)

になります．
4ビットでは，16種類のレジスタを区別可能ですが6809では，図3のように10種しか使用しません．

次に，レジスタのスタックへの退避について説明を続けます．

6809はスタック・ポインタを2本持っているので，PUSHおよびPULL命令はSP用，UP用に分かれています．

また，6800と違ってすべての内部レジスタをスタックへ退避可能としており，例のポストバイトの各ビットが図4のようにレジスタに対応しています．

PUSH，PULL命令では，1が立っているビットに対応するレジスタの内容がスタックへ退避されることになります．

図4　PUSH, PULL命令のポストバイト

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	
PC	U	Y	X	DPR	B	A	CCR	PSHS/PULS
PC	S	Y	X	DPR	B	A	CCR	PSHU/PULU

〔例3〕PSHS　A, X, PC
SPで示されるスタックにA，DPR，X，PCが退避されます．これを機械語で書くと，

アドレス		
n	$34	命令コード
n＋1	$9A	ポストバイト

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
1	0	0	1	1	0	1	0
PC	U	Y	X	DPR	B	A	CCR

このときに，スタックに積まれる順番は，図5のように決められています．

このように，わずか2バイトで任意のレジスタをスタックへ退避可能ですが，これに要するマシンサイクル数は5サイクル＋αとなり，αは退避されるバイト数×1サイクル必要です．

(6)　インデックス・アドレッシング

6809のインデックス修飾には，X，Y，UおよびSの中の1本を使うことができます．また，場合によっては，PCもインデックスと同様の働きをさせることができます．

インデックス修飾は基本的には5つに分類され，ポストバイトが，この基本タイプおよびその変形と，修飾に使われるレジスタを決定します．

図6にポストバイトのビット・パターンと，これに対応するインデックス修飾の行なわれかたを示します．

●インダイレクト・モードの例

〔例4〕インダイレクト・アドレッシング
命令実行前にYレジスタの内容は$1000とし，($1030)＝$FE，($1031)＝$50，($FE50)＝$41だったと仮定します．
LDA　〔30, Y〕
の命令を実行すると，(Y)＋$30＝$1030なので，$1030番地と，$1031番地の内容($FE50)がEA，(effective address：実効アドレス)になり，結局Aレジスタには$41がロードされます．

● b_7が"0"なら±4ビット・オフセットモードになります．6800と異なり，±のオフセットが可能です(負のオフセットは2の補数で表わす．)

● b_6およびb_5で修飾に用いるレジスタを決定．

図5　PUSH, PULL命令時，スタックに積まれる順番

PULLの順番 ↓

CCR
A
B
DPR
XHI
XLO
YHI
YLO
U/SHI
U/SLO
PCHI
PCLO

↑ PUSHの順序

メモリ・アドレスの増える方向 ↓

図6　ポストバイトとインデックス・アドレッシング・モードの種類

ポストバイト b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	インデックス・アドレッシングモード	備考	+ #	± 直接アドレス	± 間接アドレス（インダイレクト）
0	R	R	X	X	X	X	X	EA=R±4ビット・オフセット	定数オフセット（−16〜+15）	0	1	
1	R	R	0	0	0	0	0	,R+	オートインクリメント1	0	2	
1	R	R	I	0	0	0	1	,R++	オートインクリメント2	0	3	6
1	R	R	0	0	0	1	0	,−R	オートデクリメント1	0	2	
1	R	R	I	0	0	1	1	,−−R	オートデクリメント2	0	3	6
1	R	R	I	0	1	0	0	EA=,R±0オフセット	零オフセット	0	0	3
1	R	R	I	0	1	0	1	EA=,R±AccBオフセット	アキュムレータ・オフセットB	0	1	4
1	R	R	I	0	1	1	0	EA=,R±AccAオフセット	アキュムレータ・オフセットA	0	1	4
1	R	R	I	1	0	0	0	EA=R±7ビット・オフセット	定数オフセット（−128〜+127）	1	1	4
1	R	R	I	1	0	0	1	EA=,R±15ビット・オフセット	定数オフセット（−32768〜+32767）	2	4	7
1	R	R	I	1	0	1	1	EA=,R±Dオフセット	アキュムレータ・オフセットD	0	4	7
1	X	X	I	1	1	0	0	EA=,PC±7ビット・オフセット	プログラム・カウンタ・リラティブ（ショート）	1	1	4
1	X	X	I	1	1	0	1	EA=,PC±15ビット・オフセット	プログラム・カウンタ・リラティブ（ロング）	2	5	8
1	R	R	I	1	1	1	1	EA=,アドレス	エクテンデット・インダイレクト	2		5

→アドレッシングの種類（b_3〜b_0）
→インダイレクトの有無，Iが1ならインダイレクト・モード（b_7=0では，±4ビット・オフセットの符号を表わす）（b_4）
→修飾に用いるレジスタ決定（XXはdon't care）（b_6，b_5）

R	R	R：レジスタ
0	0	X
0	1	Y
1	0	U
1	1	S

→インダイレクト・モードのときのマシン・サイクル数の増加分
→ノンインダイレクト・モードのときのマシン・サイクル数の増加分
→命令バイト数の増加分

6809　命令コード一覧表

下位＼上位	0 ダイレクト	1	2 リラティブ	3	4 A	5 B	6 インデックス	7 エクステンド	8 イミディエイト	9 ダイレクト	A インデックス	B エクステンド	C イミディエイト	D ダイレクト	E インデックス	F エクステンド
0	NEG	先行バイト	BRA	LEAX	NEG				SUBA				SUBB			
1		先行バイト	BRN LBRN	LEAY					CMPA				CMPB			
2		NOP	BHI LBHI	LEAS					SBCA				SBCB			
3	COM	SYNC	BLS LBLS	LEAU	COM				SUBD CMPD CMPU				ADDD			
4	LSR		*BHS LBHS	PSHS	LSR				ANDA				ANDB			
5			**BLO LBLO	PULS					BITA				BITB			
6	ROR	LBRA	BNE LBNE	PSHU	ROR				LDA				LDB			
7	ASR	LBSR	BEQ LBEQ	PULU	ASR					STA				STB		
8	ASL LSL		BVC LBVC		ASL(LSL)				EORA				EORB			
9	ROL	DAA	BVS LBVS	RTS	ROL				ADCA				ADCB			
A	DEC	ORCC	BPL LBPL	ABX	DEC				ORA				ORB			
B			BMI LBMI	RTI					ADDA				ADDB			
C	INC	ANDCC	BGE LBGE	CWAI	INC				CMPX/CMPY/CMPS				LDD			
D	TST	SEX	BLT/ LBLT	MUL	TST				BSR	JSR				STD		
E	JMP	EXG	BGT/ LBGT				JMP		LDX/LDY				LDU/LDS			
F	CLR	TFR	BLE/ LBLE	SWI/ 2/3	CLR					STX/STY				STU/STS		

*BCC/LBCC　　**BCS/LBCS

● b_4のI（1と混同しないでください）は間接（インダイレクト）アドレッシング使用の有無を決定（Iが1なら間接アドレッシング）.

b_3〜b_0でさらに細かなアドレッシング・モードを決定.

●オートインクリメント・モードでは，命令実行の最後にレジスタの値がインクリメントされるが，オートデクリメント・モードでは，命令実行に先立ちレジスタの値がデクリメントされる．この機能により，XやYを使って，UやSと同等なスタック操作をソフトウェアで実現可能にしています.

（7）　リラティブ・アドレッシング（相対アドレッシング）

ア）　ショート/ロング・ブランチ

6800では，−128〜+127の範囲しか相対アドレスによる分岐ができなかったのが，6809には，オフセット値を示すオペランドを2バイトに拡張した**ロングブランチ命令**が追加されました.

これによって，すべてのアドレスに対し，相対分岐可能になっています.

ただし，命令数の増加にともなって，命令コードが1バイトのものと，2バイトのものがあるために，オフセット値の計算には注意が必要です.

イ）　プログラム・カウンタ・リラティブ

X，Y，U，Sと同様に，PCは実効アドレス生成の際のポインタとして使用できます.

このモードを，ア）項で述べた相対分岐命令とともに利用することで，プログラムの格納されるアドレスに依存しないプログラム（position independent programs）が作れます.

■参考引用文献

1）小原大咲：〝6809CPUボードの製作〟, I/O, '79年10月
2）土谷昌義：〝6809命令一覧表〟, I/O, '80年2月

I/O 6809 命令一覧表

アキュムレータ・論理・演算・転送命令

	A				B				D				CCR				
	Imm	Dir	Ind	Ext	Imm	Dir	Ind	Ext	Imm	Dir	Ind	Ext	5 H	3 N	2 Z	1 V	0 C
ADD	8B	9B	AB	BB	CB	DB	EB	FB	C3	D3	E3	F3	↕	↕	↕	↕	↕
ADC	89	99	A9	B9	C9	D9	E9	F9					↕	↕	↕	↕	↕
SUB	80	90	A0	B0	C0	D0	E0	F0	83	93	A3	B3	*	↕	↕	↕	↕
SBC	82	92	A2	B2	C2	D2	E2	F2					*	↕	↕	↕	↕
AND	84	94	A4	B4	C4	D4	E4	F4					•	↕	↕	0	•
OR	8A	9A	AA	BA	CA	DA	EA	FA					•	↕	↕	0	•
EOR	88	98	A8	B8	C8	D8	E8	F8					•	↕	↕	0	•
BIT	85	95	A5	B5	C5	D5	E5	F5					•	↕	↕	0	•
CMP	81	91	A1	B1	C1	D1	E1	F1	1083	1093	10A3	10B3	*	↕	↕	↕	↕
LD	86	96	A6	B6	C6	D6	E6	F6	CC	DC	EC	FC	•	↕	↕	0	•
ST		97	A7	B7		D7	E7	F7		DD	ED	FD	•	↕	↕	0	•

＊定義されていない

アキュムレータ・メモリ操作命令

	A	B	Dir	Ind	Ext	N	Z	V	C
CLR	4F	5F	0F	6F	7F	0	1	0	0
INC	4C	5C	0C	6C	7C	↕	↕	↕	•
DEC	4A	5A	0A	6A	7A	↕	↕	↕	•
COM	43	53	03	63	73	↕	↕	0	1
NEG	40	50	00	60	70	↕	↕	↕	↕
TST	4D	5D	0D	6D	7D	↕	↕	0	•
ROL	49	59	09	69	79	↕	↕	↕	↕
ROR	46	56	06	66	76	↕	↕	•	↕
ASL-LSL	48	58	08	68	78	↕	↕	↕	↕
ASR	47	57	07	67	77	↕	↕	•	↕
LSR	44	54	04	64	74	0	↕	•	↕

特殊命令

		N	Z	V	C
ABX	3A	•	•	•	•
DAA	19	↕	↕	0	↕
MUL	3D	•	↕	•	
SEX	1D	↕	↕	0	•

レジスタ間・転送交換命令

EXG	1E
TFR	1F

A	B	CCR	DPR
8	9	A	B

D	X	Y	U	S	PC
0	1	2	3	4	5

インデックス・スタックポインタ操作命令

		Imm	Dir	Ind	Ext	N	Z	V	C
LD	**X**	8E	9E	AE	BE	↕	↕	0	•
	Y	108E	109E	10AE	10BE	↕	↕	0	•
	S	10CE	10DE	10EE	10FE	↕	↕	0	•
	U	CE	DE	EE	FE	↕	↕	0	•
ST	**X**		9F	AF	BF	↕	↕	0	•
	Y		109F	10AF	10BF	↕	↕	0	•
	S		10DF	10EF	10FF	↕	↕	0	•
	U		DF	EF	FF	↕	↕	0	•

		Imm	Dir	Ind	Ext	N	Z	V	C
CP	**X**	8C	9C	AC	BC	↕	↕	↕	↕
	Y	108C	109C	10AC	10BC	↕	↕	↕	↕
	S	118C	119C	11AC	11BC	↕	↕	↕	↕
	U	1183	1193	11A3	11B3	↕	↕	↕	↕
LEA	**X**			30		•	↕	•	•
	Y			31		•	↕	•	•
	S			32		•	•	•	•
	U			33		•	•	•	•

スタック操作命令

PSH		PUL	
S	U	S	U
34	36	35	37

スタック操作時のポスト・バイト

	b7	b6	b5	b4	b3	b2	b1	b0
POST BYTE	PC	U(S)	Y	X	DPR	B	A	CCR

→ PUSH, PULL ←

ブランチ命令

ショートブランチ　2バイト命令　　ロングブランチ　3~4バイト命令

BCC・BHS	BVC	BNE	BPL	BGE	BHI	BGT	BRN	BRA
A≧M(B)		A≠M		A≧M(2')	A＞M(B)	A＞M(2')		
24	28	26	2A	2C	22	2E	21	20
1024	1028	1026	102A	102C	1022	102E	1021	16
1025	1029	1027	102B	102D	1023	102F		17
25	29	27	2B	2D	23	2F		8D
A＜M(B)		A=M		A＜M(2')	A≦M(B)	A≦M(2')		
BCS・BLO	BVS	BEQ	BMI	BLT	BLS	BLE		BSR

ジャンプ命令

	Dir	Ind	Ext
JMP	0E	6E	7E
JSR	9D	AD	BD

フラグ操作 & WAI

ANDCC	ORCC	CWAI
1C	1A	3C

その他の命令

NOP	RTI	RTS	SWI	SWI2	SWI3	SYNC
12	3B	39	3F	103F	113F	13

インデックス・アドレッシング, 間接アドレッシング時のポスト・バイト

b7	b6b5	b4	b3~b0	Indexed	Indexed Indirect
0	RR	×	×	n5, R	
1	RR	0	0	, R+	
1	RR	I	1	, R++	〔 , R++〕
1	RR	0	2	−R	
1	RR	I	3	−−R	〔 , −−R〕
1	RR	I	4	, R	〔 , R 〕
1	RR	I	5	B, R	〔B, R 〕
1	RR	I	6	A, R	〔A, R 〕
1	RR	I	8	n8, R	〔n8, R 〕
1	RR	I	9	n16, R	〔n16, R 〕
1	RR	I	B	D, R	〔D, R 〕
1	××	I	C	n8, PCR	〔n8, PCR〕
1	××	I	D	n16, PCR	〔n16, PCR〕
1	RR	I	F		〔 n16 〕

RR	Register
00	X
01	Y
10	U
11	S

FM-8のROMを解読

システム・エリアの解析

■世田谷マイコンクラブ
■COMPAC S

マシン語マニアにとって，欠かせないのが，サブルーチンの宝庫であるインタープリタの解読です．

FM-8を入手し，念のため，LEVEL 3と比較したところ，かなり類似していることがわかりました．

幸い，本誌には，LEVEL 3について，小山乃麓氏のエントリ・マップの解読と，システム・エリアの解析結果が報告されており，これらを参考にして作成した，FM-8のエントリ・マップとシステム・エリアを報告します．

FM-8 用プログラムの開発に役立ててもらえば幸いです．

なお，まだまだ未確認の部分が多くあります．気付いた点などありましたら，ぜひ編集部まで知らせてください．

◘参考文献

1）小山乃　麓："レベル3システム解析"，I/O'81年6月号，7月号

ワーク・アドレス表

FM-8 work address	
0011	1st skip stopping delimiter & temp register
0012	2nd skip stopping delimiter
0013.14	statement length+2
0015	double precision flag
0016	dimension order
0017	type of variables
0018	
0019	string area allowance flag
001A	subscript flag
001B	READ/INPUT flag
001C	tangent sign
001D	
001E.1F	next string registration pointer
0020	
0021.22	string registration pointer (0575)
0023.24	temporary storage of X (92ED)
0025.26	temporary storage of X ... return call exit (0F1F)
0027-2D	3rd FAC for multiplyer devisor
002E.30	
0031.32	temporary storage of S (7EC6)
0033.34	BASIC text top
0035.36	BASIC text end
0037.3A	
003B.3C	array variables (083E)
003D.3E	free area start (083E)
003F.40	string work area bottom (7ED2)
0041.42	actual boundary of string area (7FFE)
0043.44	current string address
0045.46	charactor area top (7FEE)
0047.48	current line number
0049.4A	continue restart line (AB07)
004B.4C	searching value
004D.4E	continue address
004F.50	current statement address
0051.52	searching data statement line number
0053.54	DATA RESTORE pointer (083B)
0055.56	READ/INPUT source pointer
0057	length of variable name
0058.59	actual address of variable ... source
005A.5B	actual address of variable ... recieve
005C	priority of term
005D	preceding variable type
005E	relatinal operator check
005F.60	block transfer destination address (006D)
0061.62	block transfer origin address
0063.64	block transfer final value (01FF)
0065.68	
0069.6A	found line address
006B.6C	address of string descripter
006D.6E	address of string variable
006F	KEY-board input flag
0070.71	KEY-board input source pointer
0072.73	saver when key input
0074	exponent of floating acumulator
0075-77	FAC (normal)
0078-7B	FAC (extend)
007C	sign part FAC
0082	auxiliary exponent (FAC2)
0083-89	auxiliary FAC (FAC2)
008A	sign part of FAC2
008B.8C	temporary register SGN/STR$/INSTR/EXP...
008D.8E	string descripter final position/temp regi (923A)
008F.90	saver of X
0091.92	spacial stack pointer
0093	FOR-NEXT/WHILE-END counter
0094	IF-ELSE counter/INSTR counter
0095	INSTR parameter
0096	INSTR parameter
0097.98	MID$ address pointer
0099	MID$ parameter
009A	MID$ parameter
009B.9C	FOR line number
009D	AUTO line number flag
009E.9F	AUTO start line number
00A0.A1	AUTO increment
00A2.A3	LIST or DELETE range start
00A4.A5	LIST or DELETE range stop
00A6.A7	current line number
00A8	FOR NEXT flag
00A9	TRACE flag
00AA.AB	FN DEFINE address
00AC	ERROR code
00AD.AE	ERROR line number
00AF.B0	stack recovery value (7ED1)
00B1	ERROR RESUME flag
00B2.B3	RESUME address
00B4.B5	ON ERROR GOTO line number
00B6.B7	SAVE of arithmetic address
00B8	PRINTUSING/ASCII code edit parameter
00B9	PRINTUSING/ASCII code edit parameter
00BA	PRINTUSING/ASCII code edit parameter
00BB.BC	PRINTUSING address
00BD	PRINTUSING parameter
00BE	PRINTUSING parameter
00BF	FILE number (channel#)
00C0	INPUT past end/device error flag
00C1	WIDTH parameter
00C2	LINE ; C8 PSET ; 1 PRESET ; 0
00C3	WIDTH
00C4	graphic semaphore
00C5.C6	LOCATE X position
00C7.C8	LOCATE Y position

00C9.CA	LINE end X position
00CB.CC	LINE end Y position
00CD	fielding constant
00CE	last position
00CF	actual head position
00D0	actual WIDTH
00D1	program protect flag
00D2-D7	GET NEXT/current ... charactor jmp (00DC.DD)
00D9	input buffer pointer
00DE	jump BIOS entry
00EB.EC	last line start position
00ED.EE	last line end position
00EF.F0	PAINT work pointer
00F1.F2	PAINT work pointer
00F3.F4	
00F5.F6	PAINT work pointer
00F7.F8	PAINT work pointer
00F9	
00FA	PAINT work data
00FB	PAINT parameter COLOR
00FC	PAINT parameter line charactor
00FD	PAINT work data
00FE	PAINT work data
0103	SYMBOL parameter
010D	LINE parameter
01A5	PSET/PRESET parameter
01AC-F	20msec timer
01D1-D3	jump to SWI 3 routine
01D4-D6	jump to SWI 2 routine
01D7-D9	jump to SWI routine
01DA-DC	jump to NMI routine (8CA3)
01DD-DF	jump to IRQ routine (D5AB)
01E0-E2	jump to FIRQ routine (CC57)
01E3.E4	LPT 0.1.2 feed code
01E5	COLOR of charactor
01E6	back color of charactor
01E7.E8	UNLIST line start
01E9.EA	cassette inter block gap lead signal count
01EB	default device number
01EC-EE	PEN sence (DEBF)
01EF	keyword number
01F0.F1	keyword table address (8D6A)
01F2.F3	statement jump table address (8FB4)
01F4	f.keyword number
01F5.F6	f.keyword table address 8F06
01F7.F8	fonctional jump table address 8CF0
01FB	default device no.
01FC.FD	keyword over
01FE-0200	
0201-0216	
0217.18	EXEC address
0219.1A	USER0 address
021B.1C	USER1 address
021D.1E	USER2 address
021F.20	USER3 address
0221.22	USER4 address
0223.24	USER5 address
0225.26	USER6 address
0227.28	USER7 address
0229.2A	USER8 address
022B.2C	USER9 address
022D-2F	DISK LOAD
0230-32	FILES from DISK
0233-35	DISK OPEN
0236-38	DISK CLOSE
0239-3B	DISK 1-charactor PUT
023C-44	
0245-47	DISK 1-charactor GET
0248-4A	DISK position
024B-4D	DISK EOF
024E-50	DISK GET
0251-53	
0254-56	WHILE statement
0257-59	WEND statement
0260-62	DISK buffer setup
0263-65	DOS GAP at C985 (main loop)
0266-68	DOS GAP at CA0E (append statement)
0269-6B	DOS GAP at AFA2 (monitor)
026C-6E	DOS GAP at 92B0 (error process)
026F-71	PEN ERROR GAP at 92B3
0272-74	APPEND TERM GAP at 968B
0275-77	keyword couvert GAP
0278-7A	PEN function GAP
027B-7D	decording GAP at C4AA
027E-80	function GAP at C7EB
0281-83	DOS GAP at 940E (disk parameter clear)
0284-86	terminal GAP at D782
0287-89	string process GAP at 965A
028A-8C	STRING regist GAP at CA85
028D-8F	DOS GAP at D8C2 (abort)
0290-92	DOS GAP at (NMI break)
0293-95	DIM GAP at 99B9

0296-98	valiable process GAP at 9C8A
0299-9B	phisical device no.
029C-A5	tabulating position
02AC.AD	f.key table pointer
02AE.AF	ON PEN GOTO table pointer (05C8)
02B0.B1	INPUT buffer pointer (05FA)
02B2.B3	FILE descripter table (06F9)
02B4.B5	FILE list pointer (06F9)
02CC.CD	INPUT buffer pointer
02CE	tape block type
02CF	tape block byte count
02D0	CASO use flag
02D1	CASO LOF data count
02D2.D3	CASO input buffer working pointer
02D4	tape file mode
02D5	check sum error flag
02D6	MOTOR off flag
02D7	MOTOR drive status
02D8	line output suppress flag
02D9	formatting suppress flag
02DA	INPUT/OUTPUT flag
02DB	FILE type
02DC	FILE mode
02DD	FILE name length
02DE-E5	FILE name
02E6	physical device number
02E7-EC	FILE option ' S8N2'00
02ED	END sentinel
02EE	LOAD EXEC flag
02EF	LOAD
02F0	LOAD M flag
02F1.F2	LOAD
02F3	LOAD
02F4.F5	clock hour
02F6.F7	clock minute
02F8.F9	clock second
02FA.FB	DATE year
02FC.FD	DATE month
02FE.FF	DATE day
0300-0A	write pen informations
030B.0C	cursor X
030C	cursor Y
030D	line number/WIDTH parameter
030E	scroll top
030F	scroll end
0310	f.key display flag
0311	formatting constant by display mode
0313	STOP key condition
0314.15	MON memory read-out address
0316.17	MON stack save
0318-1B	random number
031C-1E	RND work
031F-38	valiable 1st letter definition
0339	same as $
033A-3B	line number editting pointer
033C	DELETE flag
0343	
043D-053C	INPUT buffer
053F-0567	EDIT work area
0568-77	valiable name referrence area
0578-95	valiable name stack area
0596-97	temporary strage of 00D9.DA
0598	error flag
0599	INPUT command flag
059A	COM opened file flag
059B	TERM opened file flag
059C	TERM data (DIP swich)
059D.9E	RAM end-1
059F	RAM end flag
05C8.F7	ON PEN GOTO parameter
05F8.F9	
05FA.06F8	CAS input buffer
06F9.0706	FILE I/O routine table pointer
0719-2A	FILE OPEN status table
072B-2F	PEN ON OFF status table
0730-48	COM 0-4 subroutine jump table
0749-7A	ON KEY GOSUB jump table
077B-7F	ON KEY GOSUB
0780-9B	LINE continutity signal
0785-97	LPTO I/O routine jump table
079C-AD	COM I/O routine jump table

```
     FM-8 ROM entry map

8900   VERSION/DATE          0105  81/09/22
890A   BASIC cold start
8B68   RESET message
8BC4   COMO jump table
8BD6   BUBO jump table
8BF0   function key list
8C48   zero page area data
8CA3   NMI entry
8CDA   'Memory parity error'
8CF0-8D45  function jump table
8D46-8D69  operator jump table
8D6A-8FB3  keyword table
8FB4-922F  command routine jump table
904C-91FC   keyword-inter.lang. correspondent table
91FD   keyword A-Z jump table
9230-9241  'READY'  'BREAK'
9242  return address adjust
926D  block transfer
         0067 ; dest top
         0069 ; org top
         006D ; org end
9284  memory full test      B;word counts
9288  'memory full'
9296  X + B => X (00DE)
9297  (001C.1D) + X => X
9299  (0023.24) + X => X
        (special stack pull)
92A0  'out of memory'
92A9  'direct statement in file'
92AB  error message display with error code in Breg.
9336  error stop
9345  command start
9377  statement line saving
93E2  line number search
93F8  NEW
940E  program cancel
9414  variable
        & array cancel all val is single
9468  RESTORE
947C  END;Ready entry
9485  STOP
948D Break entry
94AC  CONT
94BE  RUN
94D2  GO
94DC  GOBUB
94FA  GOTO
9516  RETURN
9529  'undefind line No.'
9545  DATA
9548  REM & ELSE
954D  skip to colon delimiter
9550  skip to 00 delimiter
9596  IF
95AB  IF THEN or IF GOTO
95BE  ON ERROR or ON GOTO/GOSUB
9607  decimal to binaly in 004B.4C
963A  LET
9658  let string
9686  'missing operand'
968B  TERM
96EC  NOT
96F9  '&' ERR ERL USR FN & functions
9727  '(' test + formula + ')'( test
972C  ')' test
972F  '(' test
9732  comma test (00D6)
9734  compare B:00D9.DA
        if OK then get next character (00DB)
9740  'syntax error'
9745  sign change
974D  variable name check & load FAC
975F  formula + num test
9764  evaluation of formula
97B6  term operation  ... + - *
97D4  priority control
9842  priority control ( ^ )
9854  priority control (logical)
985F  priority control (MOD ¥)
9871  division (/) operator
988C  FAC push
98B5  FAC pull
98D7  logical or string
990F  charactor string
994C  AND
9951  OR
9956  XOR
995B  EQV
995D  NOT
9960  IMP
9969  DIM
9972  variable name
999E  A-Z test
        if alphbetic then carry clear
99A8  numeric test
        if numeric then carry clear
99B0  variable search
9A12  new variables
9A4E  variable finder
9A66  variable semaphore
9A78  formula + cint
9A85  array
9B00  'subscript out of range'
9B03  'illigal function'
9BCE  FRE
9BEF  D -> 0076.77 int flag set (load %)
      'duplicate definition'
9BF4  STR$
9C02  string descripter
9C3D  string registration
9C70  string spase
9D43  string concatenation
9D86  string formula evaluate
9DB6  LEN
9DBF  CHR$
9DD3  ASC
9DDE  LEFT$
9DFB  RIGHT$
9E05  MID$
9E47  GNC & then
9E49  formula to integer in B & then
9E4C  1 byte int
9E57  VAL
9E86  1-st term to X
      2-nd term to B
9EBF  formula to single.V in X
9E92  bddress integer (over 32767)
9EB3  PEEK
9EBD  POKE
9EDD  TAB(
9F0D  PRINT
9F97  CR/LF
9FAB  formatted tab
9FC0  SPC(
9FDF  lineout top addr-1 in X
        end sentinel is 00
A02F  カンジ
A0D9  INSTR
A15E  STRING$
A180  SPACE$
A196  MID$ (statement)
A230  HEX$
A231  OCT$
A29D   &O
A2AE   &H
A2D5  004B.4C -> ASC string
A2DA  List each line
A315  List/Delete range
       start 00A2.A3
       e n d 00A4.A4
A366  statement end check
A36C  current line number
       if '.' then 00A6.A7 -> 004B.4C
A37D  DEFSTR
A380  DEFINT
A383  DEFSNG
A386  DEFDBL
A3C8  DELETE
A3DF  Delete line
A3F3  AUTO
A426  TRON
A427  TROFF
A42B  USING!
A48B  USING
A5ED  FOR
A655   STEP
A688  NEXT
A732  WHILE/WEND
       & FOR/NEXT
A7B4  inner quate skip
A805-AACC  error message
AACD  ON ERROR
AAF7   ERROR
AB02  RESUME
AB42  SWAP
AB74  DEF
ABBA  illegal direct test
ABC5  FN
ACEE  VARPTR
AD05  DEF USR
AD3C  USR
AFA2  MON entry
AFDA   G commande
B007   get key in
B014   D commande
B03B   M commande
```

```
B09C   R commande
B0D7-E7  (register names)
B0F5   WHILE
B123   WEND
B17F  HARDC
B244  ANPORT
B287  real subtruction
B28D  real addition
B306  general normalizing
B385  FAC negate
B3B8  [ over flow ]
B3DC  aux register FAC-2 normalize
B3FD-B41D  constant
B41E  LOG
B465  real multiplication
B52E  load FAC-2
B58D-A7  constant
B5A8  real division
B676  load FAC by X
B6AA  FAC unload
B6F5  FAC-1 => FAC-2
B672  ! normalizing
B715  real SGN in B
B733  SGN
B73D  SGN in B
B74A  ABS
B756  real power
B7D6  FIX
B7E7  INT
B82C  numerical value
B87B  real value
B96D  int to ASC string
B978  normal real to ASCII string
BA45  real value reduction
BDCA  arithmetic operation
BDCE-BE54  constant
BE55  SQR
BE9D-4D  constant
BEC2  EXP
BF4D  RANDOMIZE
BF66  RND
BFB5-DC  (rand seed)
BFDD  CINT
C00F  string value test
C018  numerical value test
C01D  value type test (00D0)
         # ; cc
         cha ; Z
         % ; N
         ! ; V set
C02B  type convert
C037  CDBL
C04A  CSNG
C069  integer to real
C083  % normalizeing
C093  integer branching
C0A9  - operator
C0B7  + operator
C0DD  * operator
C137  sign negate
C17A  ¥ operator
C1A1  MOD operator
C1B3  ^ operator
C1CF  COS
C1D5  SIN
C221  TAN
C244-64  constant
C265  ATN
C294-B8  constant
C2B9  line input
C2DB  input prompt message & file check
C31D  input from KYBD & then
C321  'input past end'
C347  INPUT (statement)
C35D  READ
C360  input from file
C31a  LINEINPUT entry
C497-A9  'redo from start'
C4AA  intermediate code decording
C5C9  keyword translate to intermediate code
C7E0  change-case to upper-case
C7EB  function analize
C985  ststement execution
C9A5  from error resume
C9C8  each subroutine call
C9EE  spacial function
CA11  BACIC end
CA23-65  short keyword table
CA66  text end search
CA96  get next charactor
         blank eliminate
         alp,num ; C set
         endd ; Z set
```

```
CAA5  EXEC
CAB1  CLEAR
CAF1 text processing
CB03  COLOR
CB24  CONSOLE
CB8B  WIDTH
CC00  LOCATE
CC22  UNLIST
CC33  CSRLIN
CC3A  POS
CC57  FIRQ entry
CCE8  BIOS CASO jump table
CCFA  CASO EOF
CD03  CASO LOF
CD06  CASO open
CD9E  'device in use'
CDA1  'device I/O error'
CDA6  CASO close
CDBC  LOAD ?
CDBE  SKIPF
CDD9  CASO GET
CEF4  CASO PUT
CE23  CASO POS
CE8B  FILES
CEED-CF09  'searching found skip'
CF0A  files descripter
CF38  file name set entry
CFD2  'bad files descripter'
CFD7  LIST
D014  SAVE
D08C  protect test
D0CA  SAVEM
D10F  FILES close
D137  CLOSE
D145  all files CLOSE
D154  files number in B & 00AE
D16C  OPEN
D1A0  input file open
D1A3  output file open
D1A5  OPEN entry
D1B8  'bad files mode'
D1BB  'files already open'
D1CE  'device unavailable'
D1E6  run FILE
D1ED  MERGE
D1F5  LOAD
D336  INCH
D352  OUTCH
D38E  head position of actual file
D3B1  device number test
D3BE  output file check
D3C1  input file check
D3DE  EOF
D3E1  LOF
D420  INPUT
D45C  COM close
D471  COM open
D420  INPUT
D45C  COM close
D471  COM open
D513  COM PUT
D564  COM GET
D5AB  IRQ entry
D614  head position
D61D  COM EOF
D62C  COM LOF
D696  ON GOTO
D6BD  COM
D6FC  I/O status clear
D782  TERM  mode set
D865  LPT close
D870  LPT PUT
D8B8   BASIC hot start (STOP-RESET)
D8C2  ABORT test
D8F4-FD  'abort'
D8FE  close test
D9A1  MOTOR
D9C1  MOTOR ON
D9D4  MOTOR OFF
DAE8  EDIT
DB1B  one block write
DB8C-9D  BIOS SCRN
DB9E  SCRN POS
DBAA  SCRN open
DBb1  LPTO open
DBD0  SCRN PUT
DC89  CLS screen CTRL $0C
DD11  BIOS KYBD jump table
DD23  KYBD open
DD2E  BEEP
DD33  CTRL $07 (bell)
DD53  SCRN GET
DD80  INKEY$
DD90  KEY
```

I/O

エントリ・マップ

DEE3	CONNECT
DF91	SYMBOL
E01D	GCURSOR
E167	POINT
E196	PSET
E19C	PRESET
E206	line end position limitting
E228	line start POS to 00C5-C7 & POS to 00C9-CB
E256	LINE
E30C	TIME (function)
E382	DATE (function)
E3DE	TIME (statement)
E435	DATE (statement)
E4A1	clock reset
E505	INTERVAL
E624	PAINT
E6B6	CIRCLE
E953	GET
E96E	PUT
EBD2	SCREEN
EC33	BUBINI
EF6D	BUBW
EF70	BUBR
F1FE	KILL
F2D8	BIOS entry
F304	BIOS jump table
F33C	analog port input
F37A	KANJI ROM read routine
F3FE	cassete motor control
F40B	cassete 1byte wirte
F484	cassete 1byte read
F511	bubble initialize
F557	bubble write
F557	bubble read
F5CF	beep on
F5D2	beep off
F5D8	LP output
F5F5	LP check
F61F	subsystem output
F61F	subsystem input
F683	console line input
F741	input continue
F75B	console output
F7BB	key input
F7E4	HARDC1
F9A5	HARDC2
FB05	voice parameter set
FB48	voice data transfer
FB71	voice output
FB87	voice initialize
FB98	BIOS initialize
FE00	BOOT routine
FE02	seek track 0
FE05	sector write
FE08	sector read

サブルーチン表

9F97	output CR/LF
9FDF	output from X until 00
A01A	output from X for Breg. times
AFE3	output D in hex
AFE9	output A in hex
B016	hex to binary convert
B96D	output D in decimal
D336	input char in A with wait
D352	output A
D9B8	MOTOR ON
D9BB	MOTOR OFF
DA50	input line in buffer($043D),X=$043D,C=stop key
DB49	cursor blinking data set
DCFA	BIOS blinking data subout
DD69	input char in A without wait,Z=no input
F2D8	BIOS entry

FM-8活用研究

システム/アプリケーション編



FM-8

拡張版 Kコンパイラ(V1.3)

■COMPAC T & S

Kコンパイラは，整数型BASICに次の機能を拡張したものです．

- 英数字ラベルが使える．
- 引数をもったサブルーチン・コールができる．
- 再帰関数コールができる．
- REPEAT，WHILEが使える．
- IF-THEN-ELSEが使える．
- マシン語がそのままプログラム中にかける．
- 配列の割り付けを実アドレスで指定できる．

なお，コンパイル後のオブジェクトは任意の位置において実行できます(位置自由)．コンパイラ自身も位置自由です．ただし，ランタイム・ルーチンは位置固定です．また，ランタイム・ルーチンの機種に依存する部分は，1文字入出力と，1行入力サブルーチンのみです．

1. BASICとの相異点

❶ラベルと文の区切り

ＢＡＳＩＣではＧＯＴＯ文，ＧＯＳＵＢ文の飛び先を文番号で指定しますが，〝K〞には文番号の概念がなく，ラベルを使用します．ラベルは英字で始まる英数字列で，ラベル名のすぐ後にセミコロンを書くことでその場所に定義されます．

BASIC	K
10 GOTO 20 20 GOSUB 30 30 RETURN	GOTO L20 L20; GOSUB L30 L30; RETURN

1行に複数の文を書く場合のセパレータが，ＢＡＳＩＣではコロン〝：〞ですが〝K〞ではセミコロン〝；〞を使います．

BASIC	K
10 A=10:B=20	A=10;B=20

❷REPEAT，WHILE文

標準的なＢＡＳＩＣにはないＲＥＰＥＡＴ文，ＷＨＩＬＥ文が使用できます．

一般形は，

```
REPEAT
    文
UNTIL 条件式
```

```
WHILE 条件式
    文
WEND
```

です．ＲＥＰＥＡＴ文は条件が成立するまで，ＷＨＩＬＥは条件が成立している間，ＲＥＰＥＡＴ－ＵＮＴＩＬ，ＷＨＩＬＥ－ＷＥＮＤで囲まれている文を実行します．

コンパイラはこれらの文をスタックを使わないオブジェクトに展開します．

したがって，ＲＥＰＥＡＴ節，ＷＨＩＬＥ節への飛び込み，飛び出しは自由に行なうことができます．

K	展開されるコード
REPEAT I=I+1 UNTIL I>1000	L01 EQU * LDD I ADDD #1 STD I ;I=I+1 LDD I SUBD #1000 BLE L01 ; if I<=1000 then L01

K	展開されるコード
WHILE I<=1000 I=I+1 WEND	L01 LDD I SUBD #1000 LBGT L02 LDD I ADDD #1 STD I BRA L01 L02 EQU *

❸IF文

ＢＡＳＩＣのＩＦ文は1行という制約がありますが，ＫではＩＦ文の有効範囲をＦＩで明示することにより，ＴＨＥＮ節，ＥＬＳＥ節の複数文化を許しています．

ＩＦ文のネストには制限はないが，1つのＩＦに対し，必らず対応するＦＩがなくてはなりません．

BASIC	K
10 IF A>B THEN 40 20 C=B:D=A 30 GOTO 50 40 C=A:D=B 50 ⋮	IF A<=B THEN C=B;D=A ELSE C=A;D=B FI ⋮

❹GOSUB文

GOSUB 〈ラベル〉〔引数*1*, 引数*2*, …, 引数*n*〕

という形式でサブルーチンに引数を渡せます．

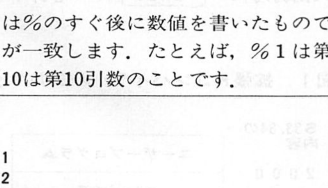
```
                        K
〔例〕   GOSUB ALPHA〔2,A, "ABCD"〕
〈展開されるコード〉
        LDD     #2
        PSHU    D
        LDD     A
        PSHU    D
        LEAX    <STR, PCR
        BRA     L01
STR     FCC     /ABCD/
        FCB     0
L01     PSHS    X
        PULU    X
        PSHS    X
        PULU    X
        PSHS    X
        LBSR    ALPHA
        LEAS    6, S
```

上の例のように引数はストリング以外は値が直接スタックに積まれ，ストリングの場合はその文字列の格納されているアドレスをスタックに積み，サブルーチンをコールします．引数は最後の1つ以外はいったんUスタックに積まれ，引数の評価がすべて終り，サブルーチンが実際にコールされる直前に，UスタックからSスタックに移されます．

このような処理を行なう理由は，スタックに積まれる順番を，引数として書かれたのと逆順にすることにより，サブルーチン側での引数参照をやりやすくしているのと，サブルーチン・コール時にSスタックのレベルを一定に保つためです．

サブルーチンが呼び出された時点でのスタックには，S，S＋1にリターン・アドレスが，S＋2，S＋3に第1引数が，S＋4，S＋5に第2引数が，一般にS＋2*n*，S＋2*n*＋1に第*n*引数が格納されています．

サブルーチンがコールされた時点でのスタックの内容



サブルーチン側で引数を参照するときは，％変数を使います．

％変数とは％のすぐ後に数値を書いたもので，数値と引数の番号とが一致します．たとえば，％1は第1引数の値であり，％10は第10引数のことです．

```
〔例〕
SUB：
    X=%1
    Y=%2
    Z=%3
    RETURN
```

というサブルーチンは，第1引数の値をXに，第2引数をYに，第3引数をZに代入します．このサブルーチンに対し，

```
    GOSUB  SUB〔3, A, B+2〕
```

という文を実行すると，Xには3が，YにはAの値が，ZにはB＋2の値が代入されます．

引数は値で渡されるので，引数の値をサブルーチン内で変化させることはできません．しかし，引数にアドレス値を渡せば，サブルーチン内でそのアドレスの内容を変化させることができます．

また，サブルーチンはサブルーチンの最後に実行された値を持ち帰る関数として使用できます．たとえば，階乗を値とするFACTは次のように記述することができます．

```
〔例〕
FACT；
    IF  %1=1 THEN A=1
             ELSE A=%1*FACT〔%1−1〕
    FI
    RETURN
```

このようなエレガントな記述は，リカーシブ・コールを許された構造化言語の特権でしたが，〝K〟では構造化言語の煩わしさなしに，それを実現しています．仮引数を名称でなくシーケンス・ナンバーで表わすことは，引数の意味を明示できなく文書性に欠けます．〝K〟のような非構造化言語では手続きの入口と出口が1つに限らず，コンパイラがローカル変数域を動的に処理することが不可能なので，このような形式をとらざるを得ませんでした．しかしながら，このような機能は標準的なBASICからは大きな進歩であると思います．

❺DEF文

ユーザーがアセンブラで記述した手続きや，システムのサブルーチンを，コールするときに，引数をSスタックを使って渡すことができます．外部ルーチンはDEF文でラベル名が与えられ，一般の手続き関数と同一に取り扱われます．

DEF　〈ラベル〉＝〈アドレス〉〔,………〕

によりラベルが指定したアドレスに定義されます．

使うときは，

〔GOSUB〕〈ラベル〉【………】

とするか，または関数として式の中で自由に使えます．関数として使うときは，ユーザーサブルーチンから戻ってきたときに，AccDに入っている値が関数の値になります．このときS，U，DPRはユーザールーチンで変化させてはいけません．

引数は一般の手続き同様，Sスタック上にpushされてユーザールーチンに渡されます．S，S＋1にリターン・アドレスが，S＋2，S＋3には第1引数の値が，S＋2n，S＋2n＋1には第*n*引数の値が入ることになります．

したがって，ユーザールーチンでは，n，Sという6809のインデックス・アドレッシング・モードを使うことにより，それを参照します．

❻配列

配列は，BASICのDIM文と異なり，ユーザーが配列の先頭アドレスを直接指定するようになっています．先頭アドレスの指定のしかたは，

変数名＝〈式〉

です．

● 2バイト配列

変数(〈式〉)……変数の内容＋〈式〉＊2のアドレスからはじまる2バイトを示す．

● 1バイト配列

変数：〈式〉）……変数の内容＋〈式〉のアドレスからはじまる1バイトを示す．

変数の値を変えることにより，配列の位置を動的に変更できます．

❼FOR，NEXT文

FORループの形式ではBASICと同一ですが，細かい点が異なります．拡張されたBASICではFORの2

重ループに対して，NEXT文で２つの変数を書くことが許されていますが，KではNEXT文で変数が指定できず，常にFORの数とNEXTの数が一致していなくてはなりません．

BASIC	K
FOR I=1 TO 100 FOR J=1 TO 100 M(I, J)=0 NEXT J, I	FOR I=1 TO 100 FOR J=1 TO 100 M (I*100+J) =0 NEXT NEXT

BASICでは初期値>最終値でもループの中を１度実行するのがほとんどですが，〝K〟ではループの中を１度も実行しません．STEPは正または負の定数が使えます．

❽PRINT文

PRINT文の形式は，BASIC同様，PRINTに続けて出力要素の並びを書きます．BASICでは要素を出力した後には改行するのが基本ですが，〝K〟では改行は行ないません．PRINT文において改行したいときはFORTRANよろしく，スラッシュ〝/〟を書きます．

BASIC	K
PRINT〝ABC〟	PRINT〝ABC〟/
PRINT〝12,3〟;	PRINT〝123〟

出力要素はコンマを使って複数個並べることができます．

BASIC	K
PRINT A;B	PRINT A, B, /
PRINT A;〝123〟;	PRINT A, 〝123〟

上の例で，Bと〝/〟の間にコンマを入れる必要があることに注意してください．もし，コンマがないと，コンパイラは〝/〟を改行子としては解釈せず，除算の演算子と解釈し，ILLEGAL EXPRESSION ERRORとします．

BASICのコンマによるタブレーションは〝K〟にはないが，それに代わる機能として右詰め出力というのがあります．

#<式１>，<式２>

というフォーマットで<式２>の値を<式１>の桁数をとり右に詰めて出力します．

BASIC	K
PRINT USING〝####〟;A;	PRINT #4, A

また，この他にPRINT文中でHEX$(<式>)，HEX2$(<式>)，CHR$(<式>)，SPC$(<式>) が使用可能です．

❾INPUT，GET文

BASICではINPUT，GETは文ですが，〝K〟ではキーボードから入力した値，キーのASCIIコードを値とする関数です．

BASIC	K
INPUT A, B	A=INPUT ; B=INPUT
INPUT〝C=〟; C	PRINT〝C=〟; C=INPUT
GET A$: A=ASC(A$)	A=GET

2．使い方

2-1　プログラムの作成

〝K〟のプログラムは，BASICのエディタを使って，〝'〟（注釈）の形式で作成します．行の最初が〝'〟でないと，コンパイル・エラーとなります．

プログラムの中では，英大小文字は文字列（' '，" "）を除いて，区別しません．ですから，キーワードを英小文字にすることもできます．

例　10 'PRINT" a="

作成したプログラムは，通常のBASICプログラムと同じく，SAVEコマンドにより，テープまたはディスクにセーブします．一度セーブしたプログラムは，LOADコマンドにより，ロードして使えます．

プログラム領域は＄１FFFまでなので，プログラム作成前に，CLEAR 10, & H1FFFとしておけば，コンパイラを壊すことがありません．

2-2　コンパイラ

コンパイラとランタイム・ルーチンがメモリ上にあることを確認して（もしなければロードする），BASICモードから，

EXEC & H2000

とします．

コンパイル・エラーを生じたときは，エラーになった文番号を表示するので，エディタに戻ってエラーを修正します．

2-3　実行

コンパイルが正常に終ったことを確認して，BASICモードから，

EXEC & H5000

とします．

実行時のGET文，INPUT文，OUTPUT文による入出力は，特に指定しなければ，キーボードからの入力，画面への出力となっています．

GET文，INPUT文による入力は，ファイル番号１に対応するので，BASICモードから，

OPEN "I", 1, "入力機器"

と指定することにより，任意の媒体から入力ができます．

PRINT文による出力は，ファイル番号２に対応するので，BASICモードから，

OPEN "O", 2, "出力機器"

と指定することにより，任意の媒体に出力ができます．

実行中 **STOP** キーによる割り込みを受け取るには，次のようにします．

アドレス	旧	新
＄1E1	＄CC	＄4F
＄1E2	＄57	＄30

これで，**STOP** キーが押されるたびに，＄４F34の値が１ずつ繰り上がります．＄４F34の値を調べれば，実行を中止することもできます．

なお，入出力時に **STOP** キーを押すと，実行を中止します．

図１．拡張Kコンパイラ メモリ・マップ

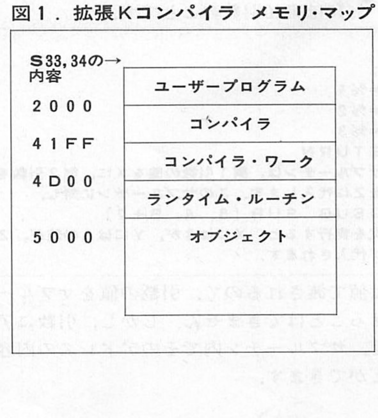


表1　拡張Kコンパイラ・エントリ・マップ

アドレス	内　容	機　　　能
2001	4C	コンパイル時作業エリア・アドレスの上位2バイト
20C0	4C	実行時変数エリア・アドレスの上位2バイト
20CE	4C	実行時Uスタック・エリア最終アドレスの上位2バイト
2073,4	4300	プログラム1行入力バッファ・アドレス
207DE	5000	オブジェクト作成アドレス
2082,3	4400	ラベル・テーブル・アドレス
208E,F	4AFE	コンパイル時スタック・エリアの最終アドレス
209A,B	4B00	作業用テーブル・アドレス
4186	サブルーチン	コンパイラ1行入力ルーチン
41E2	〃	コンパイラOPENルーチン
41EE	〃	コンパイラCLOSEルーチン
4F80	〃	ランタイム1文字入力ルーチン．エコーバックなし．
4F9B	〃	ランタイム1文字出力ルーチン．
4FB5	〃	ランタイム1行入力ルーチン，エコーバックあり．バック・スペースあり．

表2　エラーメッセージ

DOUBLE DEF ERROR	ラベルの二重定義がある．
ELSE WITHOUT IF	IF文がないのに，ELSEがきた．
FI WITHOUT IF	IF文がないのに，FIがきた．
ILLEGAL ARGUMENT	手続，関数のパラメータに誤りがある．
ILLEGAL EXPRESSION	許されない式である．
ILLEGAL NESTING	FOR, IF, REPEAT, WHILE 句の対応がおかしい．
NEXT WITHOUT FOR	FOR文がないのに，NEXTがきた．
SYNTAX ERROR	構文に誤りがある．
TOO MANY VARIABLES	変数の数が多すぎる．
UNDEFINED LABELS	未定義のラベルである．
UNTIL WITHOUT REPEAT	REPEAT文がないのに，UNTILがきた．
WEND WITHOUT WHILE	WHILE文がないのに，WENDがきた．

3．ファイル版への変更

Kコンパイラのアドレス＄４１８６～＄４１DDを図２のように変更すると，オン・メモリ版からファイル版に変更できます．

プログラムをファイルにセーブする際は，ASCIIセーブしてください．その後，図３のプログラムでKコンパイラを走らせることができます．

図２．ファイル版変更箇所

```
00100                              NAM   IO
00110  4186:                       ORG   $4186
00120  4186 34   48          GETL  PSHS  DP,U
00150  4188 BD   4FB5              JSR   $4FB5
00151  418B 86   0D                LDA   #$0D
00152  418D 109E 00                LDY   $00
00153  4190 A7   A4                STA   ,Y
00160  4192 108E 0000              LDY   #$0
00170  4196 EC   84                LDD   ,X
00171  4198 1083 7F0D              CMPD  #$7F0D
00172  419C 27   28   41C6         BEQ   LOOP
00180  419E A6   80          SKIP  LDA   ,X+
00190  41A0 81   0D                CMPA  #$0D
00200  41A2 27   31   41D5         BEQ   RET
00210  41A4 81   27                CMPA  #$27
00220  41A6 27   1E   41C6         BEQ   LOOP
00230  41A8 80   30                SUBA  #$30
00240  41AA 2B   F2   419E         BMI   SKIP
00250  41AC 81   39                CMPA  #$39
00260  41AE 22   EE   419E         BHI   SKIP
00270  41B0 34   02                PSHS  A
00280  41B2 1F   20                TFR   Y,D
00290  41B4 34   06                PSHS  D
00300  41B6 58                     ASLB
00310  41B7 49                     ROLA
00320  41B8 58                     ASLB
00330  41B9 49                     ROLA
00340  41BA E3   E1                ADDD  ,S++
00350  41BC 58                     ASLB
00360  41BD 49                     ROLA
00370  41BE EB   E0                ADDB  ,S+
00380  41C0 89   00                ADCA  #0
00390  41C2 1F   02                TFR   D,Y
00400  41C4 20   D8   419E         BRA   SKIP
00401  41C6 34   20          LOOP  PSHS  Y
00402  41C8 109E 00                LDY   $00
00410  41CB A6   80          LOOP1 LDA   ,X+
00420  41CD A7   A0                STA   ,Y+
00430  41CF 81   0D                CMPA  #$0D
00440  41D1 26   F8   41CB         BNE   LOOP1
00450  41D3 35   20                PULS  Y
00460  41D5 109F 12          RET   STY   $12
00470  41D8 9E   00                LDX   $00
00480  41DA 9F   16                STX   $16
00490  41DC 35   C8                PULS  DP,U,PC
00500                              END
```

図３．ファイル版メイン・プログラム

```
10 CLEAR100,&H6FFF
11 INPUT"HOT OR COLD";A$
12 IF A$="H" THEN 40
20 LOADM"RT
30 LOADM"KF
40 INPUT"FILE NAME";A$
50 OPEN"I",#1,A$
55 IF PEEK(&H2DC)=0 THEN PRINT"** NOT ASCII FILE **":CLOSE #1:END
56 PRINT"FILE IS OPENED"
60 EXEC&H2000
70 CLOSE #1
80 CLEAR300,&H4B00
```

4．デバッグ

KコンパイラV1.1には次のバグがあるので，訂正願います．数値は16進数です．V1.3はV1.2のランタイム・ルーチンの1行入力ルーチン＄4FB5～＄4FE0がファイル入力用に変更されています．

アドレス	旧内容	新内容
2830	E8	E9
3BD9	CC	EC

5．プログラム例

〔例1〕8クイーン

有名な8クイーンの問題です．すなわち，8個のクイーンを互いに他をとることができないようにチェス盤の上に配置する問題です．

このプログラムは，すべての解（対称性は無視して92個）を求めます．手続き QUEEN は QUEEN の中で QUEEN を（再帰的に）呼んでます．%変数はローカルな変数となり，スタックに積まれるので，再帰的に使えます．

STOP キーを押すと処理を中止するようにしています．CONST文により，%変数に意味を持たせています．

8クイーン・プログラム

```
100 ' (* STOP KEY INT. SET *)
110 ' CONST STOP=$4F34
120 ' POKE $1E1,$4F;POKE $1E2,$30;POKE $4F34,0
130 ' (* 8 QUEEN MAIN *)
140 ' SIZE=8;BOARD=$6000
150 ' K=0;QUEEN[1,DUMMYN,DUMMY]
```

8クイーン・プログラム

```
160 'END
170 'PR; (* PRINT ONE SOLUTION *)
180 ' K=K+1;PRINT #3,K,":"
190 ' FOR I=1 TO SIZE
200 '   PRINT #2,BOARD(I)
210 ' NEXT
220 ' PRINT /
230 ' IF PEEK(STOP) THEN END FI
240 'RETURN
250 'QUEEN; (* GET ALL 8 QUEENS *)
260 ' CONST L=1,N=2,J=3
270 ' IF %L>SIZE THEN GOSUB PR
280 '   ELSE %N=1
290 '     WHILE %N <=SIZE
300 '       %J=1
310 '       WHILE %J<=%L-1
320 '         IF %N=BOARD(%J) OR %N+%L=BOARD(%J)+%J OR %N-%L=BOARD(%J)-%J THEN
330 '             GOTO L1
340 '         FI
350 '         %J=%J+1
360 '       WEND
370 '         BOARD(%L)=%N
380 '             QUEEN[%L+1,DUMMYN,DUMMYJ]
390 '         L1;%N=%N+1
400 '     WEND
410 ' FI
420 'RETURN
```

〔例2〕高速ソート

最良のソート法といわれるクイック・ソートと，小データに対してバブル・ソートを使った高速ソート・プログラムです．1024個のランダムなデータを，2秒弱でソートします．

手続きQUICKは，QUICKの中でQUICKを呼んでいます．手続きEXCHANGEは，引数にアドレス値を持ち，手続きの中でそのアドレスの内容を直接変更しています．

高速ソート・プログラム

```
100 '(* HIGH SPEED SORTING *)
110 ' DATA=$6000
120 ' PRINT"HOW MANY DATA?";N=INPUT
130 ' FOR I=1 TO N
140 ' (* PRINT"DATA[",I,"]=";DATA(I)=INPUT *)
150 '    DATA(I)=RND(1000)
160 ' NEXT
170 ' PRINT"NOW SORTING..."/
180 ' QUICK[1,N]
190 ' FOR I=1 TO N
200 '    PRINT"DATA[",I,"]=",DATA(I),/
210 ' NEXT
220 'END
230 'QUICK; (* QUICK SORT FOR LARGE DATA *)
240 '   IF %2-%1<8 THEN BUBBLE[%1,%2]
250 '   ELSE
260 '      QUMIDDLE=(%1+%2)/2
270 '      QUM=MID[DATA(%1),DATA(QUMIDDLE),DATA(%2)]
280 '      QUF=%1;QUL=%2
290 '      WHILE QUF<=QUL
300 '         WHILE DATA(QUF)<QUM;QUF=QUF+1;WEND
310 '         WHILE DATA(QUL)>QUM;QUL=QUL-1;WEND
320 '         IF QUF<=QUL THEN
330 '            WORK=DATA(QUF);DATA(QUF)=DATA(QUL);DATA(QUL)=WORK
340 '            QUF=QUF+1;QUL=QUL-1
350 '         FI
360 '      WEND
370 '      QUICK[%1,QUL]
380 '      QUICK[QUF,%2]
390 '   FI
400 'RETURN
410 'BUBBLE; (* BUBBLE SORT FOR SMALL DATA *)
420 '   BUBOUND=%1
430 '   WHILE BUBOUND<%2
440 '      BULASTEX=%2-1
450 '      FOR BUI=%2-1 TO BUBOUND STEP -1
460 '         IF DATA(BUI)>DATA(BUI+1) THEN
470 '            WORK=DATA(BUI);DATA(BUI)=DATA(BUI+1);DATA(BUI+1)=WORK
```

```
480 '              BULASTEX=BUI
490 '           FI
500 '        NEXT
510 '        BUBOUND=BULASTEX+1
520 '     WEND
530 'RETURN
540 'MID; (* GET MIDDLE VALUE *)
550 '     MIDWRK1=%1;MIDWRK2=%2;MIDWRK3=%3
560 '     IF %2>%3 THEN EXCHANGE[ADR(MIDWRK2),ADR(MIDWK3)];FI
570 '     IF %1>%2 THEN EXCHANGE[ADR(MIDWRK1),ADR(MIDWRK2)];FI
580 '     IF %2>%3 THEN EXCHANGE[ADR(MIDWRK2),ADR(MIDWRK3)];FI
590 'MID=MIDWRK2
600 'RETURN
610 'EXCHANGE;(* EXCHANGE DATA *)
620 '     EX1=%1;EX2=%2;EXTEMP=EX1(0);EX1(0)=EX2(0);EX2(0)=EXTEMP
630 'RETURN
```

〔例3〕シェルピンスキー曲線

シェルピンスキー曲線と呼ばれる，再帰的曲線を描きます．手続きA，B，C，Dはそれぞれ再帰的な手続きで，図形の4つの部分を90°ずつ回転して描きます．

最初に，何位まで描くか指定します．位数により色を変えて描きます．

手続きPLOTは，FM-8のサブCPUに直接コマンドを与えて，線分を描きます．

位数4のとき，約6秒で全曲線を描きます．

シェルピンスキー　プログラム

```
100 '(* SHIERPINSKI CURVE PLOT OF ORDER N *)
110 'CONST HO=128
115 'N=INPUT
120 ' H=HO/4;XO=4*H;YO=3*H
130 ' PRINT CHR$(12);(* CLEAR SCREEN *)
140 ' I=0;COLOR=1
150 'REPEAT;
160 '   I=I+1;XO=XO-H
170 '   H=H/2;YO=YO+H
180 '   X=XO;Y=YO
190 '   XOLD=X*2;YOLD=Y
200 '   A[I];X=X+H;Y=Y-H;PLOT[X*2,Y]
210 '   B[I];X=X-H;Y=Y-H;PLOT[X*2,Y]
220 '   C[I];X=X-H;Y=Y+H;PLOT[X*2,Y]
230 '   D[I];X=X+H;Y=Y+H;PLOT[X*2,Y];COLOR=COLOR+1
240 'UNTIL I=N
250 'END
260 'A;IF %1>0 THEN
270 '   A[%1-1];X=X+H;Y=Y-H;PLOT[X*2,Y]
280 '   B[%1-1];X=X+2*H;PLOT[X*2,Y]
290 '   D[%1-1];X=X+H;Y=Y+H;PLOT[X*2,Y]
300 '   A[%1-1];FI
310 'RETURN
320 'B;IF %1>0 THEN
330 '   B[%1-1];X=X-H;Y=Y-H;PLOT[X*2,Y]
340 '   C[%1-1];Y=Y-2*H;PLOT[X*2,Y]
350 '   A[%1-1];X=X+H;Y=Y-H;PLOT[X*2,Y]
360 '   B[%1-1];FI
370 'RETURN
380 'C;IF %1>0 THEN
390 '   C[%1-1];X=X-H;Y=Y+H;PLOT[X*2,Y]
400 '   D[%1-1];X=X-2*H;PLOT[X*2,Y]
410 '   B[%1-1];X=X-H;Y=Y-H;PLOT[X*2,Y]
420 '   C[%1-1];FI
430 'RETURN
440 'D;IF %1>0 THEN
450 '   D[%1-1];X=X+H;Y=Y+H;PLOT[X*2,Y]
460 '   A[%1-1];Y=Y+2*H;PLOT[X*2,Y]
470 '   C[%1-1];X=X-H;Y=Y+H;PLOT[X*2,Y]
480 '   D[%1-1];FI
490 'RETURN
500 'PLOT; (* PLOT[X,Y] *)
540 ' A=$FC82
551 ' CODE $B6,$FD,$05,$2B,$FB; (* READY ? *)
552 ' POKE $FD05,$80; (* HALT SUB CPU *)
553 ' CODE $B6,$FD,$05,$2A,$FB
560 ' CPOKE A,22,7,0,2,XOLD#,YOLD#,%1#,%2#
570 ' POKE $FC83,COLOR;(* PLOT LINE *)
580 ' POKE $FD05,0; (* RESTART SUB CPU *)
581 ' XOLD=%1;YOLD=%2
590 'RETURN
```

Kサブ・プログラム

```
100 'A=$6000
110 'A=A(0)
```

Kサブ・プログラム

```
120 'IF PEEK($6002) THEN GOTO PUT ;FI
130 ' FOR I=0 TO 2;PRINT "DATA",I,"=",A(I),/;NEXT
140 ' FOR I=6 TO 10;PRINT CHR$(A:I));IF A:I)=$OD THEN GOTO RET;FI;NEXT
150 ' GOTO RET
160 'PUT;
170 ' FOR I=0 TO 2;PRINT "DATA",I,"=";A(I)=INPUT;NEXT
180 ' PRINT "NAME="
190 ' FOR I=6 TO 10;A:I)=GET;PRINT CHR$(A:I));IF A:I)=$OD THEN GOTO RET;FI;NEXT
200 'RET;PRINT /;CODE $35,$FF
```

〔例4〕ランダム・ファイル

F-BASICのランダム・ファイルに，データを数値のまま(文字に変換せず）入れておきたい場合があります．

このプログラムは，BASICのメイン・プログラムがランダム・ファイルの入出力を行ない，Kのサブルーチンがキーボードからの入力とメモリ上のバッファとの入出力を行なう例です．

BASICとKでは，DP（ダイナミック・ページ）レジスタの値が違うので注意が必要です．

Kで作成された機械語プログラムの先頭には，**NOP**（$12）命令が2個あるので，それを＄34，＄7F（**PSHS**）に変更します．Kの中でBASICに戻るには，**RETURN**文でなく，**CODE**文により，＄35，＄FF（**PULS**）で戻ります．

CLEAR文によるメモリ上限は，＄4B00程度までにしてください．

◆参考文献

1)I/O，82年1月号マイコン大学
2)K.L.Bowles，"UCSD PASCAL演習"，工学社
3)N.Wirth，"アルゴリズム＋データ構造＝プログラム"，日本コンピュータ協会
4)FM-8，ユーザーズマニュアルF-BASIC入門

BASIC メイン・プログラム

```
100 CLEAR300,&H4B00
110 OPEN"R",#3,"RFILE"
120 FIELD #3,255 AS A$
130 LSET A$=STRING$(255,&H00)
140 POKE &H5000,&H34:POKE &H5001,&H7F
150 POKE &H6000,PEEK(VARPTR(A$)+1)
160 POKE &H6001,PEEK(VARPTR(A$)+2)
170 INPUT"U(PDATE),R(EAD),E(ND)";ANS$
180 IF ANS$="E" THEN CLOSE #3:END
190 IF ANS$<>"R" THEN GOTO 250
200 INPUT"RECORD NO.=",RNO
210 IF RNO>LOF(3) THEN PRINT"END OF FILE":GOTO 170
220 GET #3,RNO
230 POKE &H6002,0
240 GOTO 280
250 IF ANS$="U" THEN POKE &H6002,1:GOTO 270
260 PRINT CHR$(7),"COMMAND ERROR":GOTO 170
270 INPUT"RECORD NO.=",RNO
280 EXEC&H5000
290 IF ANS$="U" THEN PUT #3,RNO
300 GOTO 170
```

Kコンパイラ言語仕様（V1.2）

数値	16ビット符号付き2進数 10進数　−32768〜32767 16進数　$0000 〜$FFFF 1文字定数　'A'，'B' ……	
変数	単純変数	英字で始まる英数字列で長さは任意だが8文字まで識別する． 例　LOVELYRUMICHAN MIURA27
	配列	単純変数の後に(〈式〉)または：〈式〉)を付けたもの． 例　HIZENBERG（DIRAC） QUARK：COLOR)
	%変数	%の後に数値を書いたもの．引数の参照に用いる． A＝%1 B＝%2
関数	ラベル名の後に大カッコで囲った引数を書いたもの． 例　A＝LEN〔"ABCD"〕＋3	
ラベル	単純変数と同じ．	

関数

一般形	使用例	機能
PEEK(〈アドレス〉)	PEEK($1000)	アドレスのメモリ内容を値とする
RND(〈式〉)	RND(100)	0〜式の値−1の一様乱数を発生 0≦RND(n)＜n
INPUT		キーボードより入力する数値を値とする
GET		キーが押されるまで待ち，押されたキーのASCIIコードを値とする
NOT(〈式〉)	NOT(A＞1)	論理否定演算
ABS(〈式〉)	ABS(A＋B)	絶対値
−〈式〉	−100	負数
LOW(〈式〉)		式の下位バイトを値とする
HIGH(〈式〉)		式の上位バイトを値とする
ADR(変数)		変数のアドレスを値とする
SGN(〈式〉)		式の符号を値とする．0は0を値とする．
SEX(〈式〉)		1バイトの負の数を2バイトの負の数にする．

文

一般形	使用例	機能
GOTO〈ラベル〉	GOTO ALPHA	ラベルにJUMP
GOSUB〈ラベル〉	GOSUB DIRAC	ラベルをサブルーチン・コール
RETURN	RETURN	サブルーチンよりリターン
REPEAT 文 UNTIL〈条件式〉	REPEAT I＝I＋1 UNTIL I＞1000	条件が成立するまでループ
WHILE〈条件式〉 文 WEND	WHILEI＜1000 I＝I＋1 WEND	条件が成立している間ループ
FOR〈変数〉＝〈初期値〉TO〈最終値〉〔STEP〈きざみ値〉〕 文 NEXT		繰り返し
IF〈条件式〉THEN 文1 ELSE 文2 FI	IF A＝'A' THEN B＝'B'；C＝'C' ELSE B＝'D'；C＝'E' FI	条件が成立していると文1を成立しないときは文2を実行する．
CALL〈アドレス〉	CALL $F000	ユーザーサブルーチン，システム・サブルーチンのコール
POKE〈アドレス〉，〈値〉	POKE $200,B＋3	アドレスに値をストア
PRINT 出力要素〔，…〕	PRINT "ABC",A	CRTに対するプリント・アウト
CODE〈値〉〔，…〕	CODE $34, $1C, $17	値をプログラムのこの文の位置に展開
CPOKE〈変数〉，〈式〉〔，…〕	CPOKE A, $34, $08	Aのアドレスから連続的にPOKEする．〈式〉の後に#があると，2バイトにして書き込む．
DEF〈ラベル〉＝アドレス〔，〈ラベル〉…〕	DEF ML＝$5000 CALL ML	ラベルにアドレスを定義する．
CONST〈変数〉＝〈値〉〔，〈変数〉…〕	CONST N＝1 A＝%N	コンパイル時，指定した変数が現れたら数値に置き換える．
END		実行を終了する．

コメント

一般形	使用例	機能
(＊　＊)	(＊REMARK＊)	注釈

ＰＲＩＮＴ文中で使用できる関数

一般形	使用例	機能
/		改行子
CHR$(<式>)	CHR$(4)	式の値をASCIIコードとして出力
HEX$(<式>)	HEX$(−121)	式の値を16進4桁で出力
HEX2$(<式>)	HEX2$("A")	式の値を16進2桁で出力
SPC$(<式>)	SPC$(3)	式の値の数のスペースを出力
"<文字列>"	"HITOMI"	" "で囲まれた部分を出力
#<式1>, <式2>	#4, A	<式1>のスペースをとって<式2>を右ずめ出力.

演算子

一般形	使用例	機能
*	X=2*3	乗算演算子
／	X=Y/3	除 〃
＋	X=X+1	加 〃
−	X=X−1	減 〃
＞	X>1	関係演算子
＞＝	X>=1	
＜	X<1	
＜=	X<=1	
＝	X=1	
＜＞	X<>1	
AND	X=1 AND X=2	論理演算子
OR	X=1 OR X=2	

優先順位は乗除，加減，関係，論理の順

拡張Kコンパイラ　ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2000 C6 4C 1F 9B CE 44 00 8D 07 C6 00 1F 9B 7E 94 7C :80
2010 BD 4D 17 BD 4D 0B 4B 20 43 4F 4D 50 49 4C 45 52 :FC
2020 20 56 45 52 53 49 4F 4E 20 31 2E 32 00 BD 4D 17 :18
2030 BD 4D 0B 43 4F 50 59 52 49 47 48 54 20 31 39 38 :90
2040 32 20 42 59 20 43 4F 4D 50 41 43 2E 00 BD 4D 17 :0F
2050 BD 4D 17 BD 4D 17 BD 4D 0B 43 4F 4D 50 49 4C 49 :64
2060 4E 47 2E 2E 2E 00 BD 4D 17 BD 4D 17 BD 4D 17 17 :99
2070 21 70 CC 43 00 DD 00 CC 00 01 DD 02 CC 50 00 DD :22
2080 04 CC 44 00 DD 06 CC 00 FF 9E 06 E7 84 CC 4A FE :E5
2090 DD 08 CC FF 00 9E 08 ED 84 CC 4B 00 DD 0A CC 00 :91
20A0 00 DD 0C CC 00 0D DD 0E CC 00 7F DD 10 CC 00 00 :B1
20B0 DD 12 CC 00 00 DD 14 CC 12 12 9E 04 ED 84 CC C6 :41
20C0 4C 9E 04 ED 02 CC 1F 9B 9E 04 ED 04 CC CE 4C 9E :7A
20D0 04 ED 06 CC 00 00 9E 04 E7 08 DC 04 C3 00 09 DD :DD
20E0 04 17 20 A2 9E 16 4F E6 84 93 10 10 26 01 9D 17 :D8
20F0 20 FC CC C6 00 9E 04 ED 84 CC 1F 9B 9E 04 ED 02 :D8
---------------------------------------------------------
Sum: F0 C1 B7 60 D5 2D 91 39 13 B6 E5 04 8E 54 D0 C9 :C1
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2100 CC 7E 94 9E 04 ED 04 CC 00 8D 9E 04 E7 06 DC 04 :39
2110 C3 00 07 DD 04 9E 08 EC 84 83 FF 00 10 27 00 94 :0E
2120 17 20 CC BD 4D 17 BD 4D 0B 49 4C 4C 45 47 41 4C :33
2130 20 4E 45 53 54 49 4E 47 2E 00 BD 4D 17 BD 4D 0B :9C
2140 4C 49 4E 45 00 CC 00 06 36 06 9E 08 EC 08 BD 4E :DB
2150 8A 9E 08 EC 84 83 FF FF 10 26 00 08 BD 4D 0B 20 :94
2160 46 4F 52 00 9E 08 EC 84 83 FF FE 10 26 00 07 BD :77
2170 4D 0B 20 49 46 00 9E 08 EC 84 83 FF FD 10 26 00 :D2
2180 0B BD 4D 0B 20 52 45 50 45 41 54 00 9E 08 EC 84 :17
2190 83 FF FC 10 26 00 0A BD 4D 0B 20 57 48 49 4C 45 :6C
21A0 00 BD 4D 17 DC 08 C3 00 0A DD 08 9E 08 EC 84 83 :50
21B0 FF 00 26 89 DC 06 DD 18 CC 00 00 DD 1A 9E 18 4F :4D
21C0 E6 84 83 00 FF 10 27 00 6D 9E 18 EC 84 83 00 01 :3A
21D0 10 26 00 58 DC 1A 83 00 00 10 26 00 1F 17 20 0F :A2
21E0 BD 4D 17 BD 4D 0B 55 4E 44 45 46 49 4E 45 44 20 :E8
21F0 4C 41 42 45 4C 53 20 3A 00 BD 4D 17 BD 4D 0B 20 :63
---------------------------------------------------------
Sum: BB DE 0C 1A 83 2A AE 8A 8B E1 12 DA D5 9D A2 05 :15
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2200 00 CC 00 04 DD 1C CC 00 0B 36 06 DC 1C 16 00 11 :FB
2210 DC 1C 9E 18 E6 8B 4F BD 4E DE DC 1C C3 00 01 DD :F0
2220 1C A3 C4 2F EB 33 42 CC 00 01 DD 1A DC 18 C3 00 :8D
2230 0C DD 18 16 FF 87 BD 4D 17 BD 4D 17 BD 4D 0B 4F :43
2240 42 4A 45 43 54 20 50 52 4F 47 52 41 4D 20 53 49 :5C
2250 5A 45 20 3A 00 DC 04 83 4F FF DD 1E DC 1E BD 4E :AA
2260 B0 BD 4D 0B 20 28 24 00 DC 1E BD 4E BF BD 4D 0B :0A
2270 29 00 BD 4D 17 BD 4D 17 BD 4D 0B 43 4F 4D 50 49 :F8
2280 4C 45 20 45 4E 44 2E 00 BD 4D 17 39 17 12 33 9E :0A
2290 16 4F E6 84 36 06 CC 00 3B BD 4D B9 36 06 9E 16 :C5
22A0 4F E6 84 36 06 CC 00 3B BD 4D B9 AA C0 EA C0 10 :E3
22B0 27 00 0A DC 16 C3 00 01 DD 16 16 FF CF 9E 16 4F :C1
22C0 E6 84 93 0E 10 26 00 03 16 FE 16 DC 14 36 06 CC :66
22D0 00 01 BD 4D B9 36 06 30 8C 02 20 03 28 2A 00 34 :67
22E0 10 17 10 EE 32 62 AA C0 EA C0 10 27 00 5D CC 00 :2D
22F0 01 DD 14 DC 16 C3 00 01 DD 16 9E 16 4F E6 1F 36 :D9
---------------------------------------------------------
Sum: 48 A7 F1 36 E9 9C 89 F2 A2 C6 1A D0 16 06 14 71 :09
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2300 06 CC 00 2A BD 4D B9 36 06 9E 16 4F E6 84 36 06 :A4
2310 CC 00 29 BD 4D B9 A4 C0 E4 C0 36 06 9E 16 4F E6 :E5
2320 1F 36 06 DC 0E BD 4D B9 AA C0 EA C0 27 C5 9E 16 :BC
2330 4F E6 1F 93 0E 10 26 00 03 16 FD A5 DC 16 C3 00 :9B
2340 01 DD 16 CC 00 00 DD 14 16 FF 41 30 8C 02 20 06 :EB
2350 50 52 49 4E 54 00 34 10 17 10 77 32 62 36 06 30 :6F
2360 8C 02 20 02 3F 00 34 10 17 10 67 32 62 AA C0 EA :A9
2370 C0 10 27 01 BF 17 11 4A 9E 16 4F E6 84 83 00 2C :45
2380 10 26 00 0A DC 16 C3 00 01 DD 16 16 FF E7 9E 16 :99
2390 4F E6 84 83 00 2F 10 26 00 14 CC 4D 17 34 06 17 :36
23A0 10 84 32 62 DC 16 C3 00 01 DD 16 16 FF C7 9E 16 :61
23B0 4F E6 84 83 00 23 10 26 00 3E DC 16 C3 00 01 DD :66
23C0 16 17 11 75 9E 16 4F E6 84 83 00 2C 10 27 00 03 :09
23D0 16 1C 5B CC 36 06 9E 04 ED 84 DC 04 C3 00 02 DD :2A
23E0 04 DC 16 C3 00 01 DD 16 17 11 4E CC 4E 8A 34 06 :01
23F0 17 10 33 32 62 16 FF 7D 9E 16 4F E6 84 83 00 22 :92
---------------------------------------------------------
Sum: E2 BE E3 1B 66 9B 95 F6 A1 A3 EE A5 D8 F0 45 76 :84
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2400 10 26 00 61 CC 4D 0B 34 06 17 10 1A 32 62 DC 16 :BC
2410 C3 00 01 DD 16 9E 16 4F E6 84 9E 04 E7 84 DC 04 :11
2420 C3 00 01 DD 04 9E 16 4F E6 84 36 06 CC 00 22 BD :F9
2430 4D B9 36 06 9E 16 4F E6 84 36 06 DC 0E BD 4D B9 :98
2440 AA C0 EA C0 27 C8 CC 00 00 9E 04 E7 1F 9E 16 4F :7A
2450 E6 84 93 0E 10 26 00 03 16 FC 86 DC 16 C3 00 01 :92
2460 DD 16 16 FF 10 30 8C 02 20 06 43 48 52 24 28 00 :25
2470 34 10 17 0F 5D 32 62 10 27 00 10 17 10 9F CC 4E :82
2480 DE 34 06 17 0F A0 32 62 16 FE EA 30 8C 02 20 06 :54
2490 48 45 58 24 28 00 34 10 17 0F 37 32 62 10 27 00 :9D
24A0 10 17 10 79 CC 4E BF 34 06 17 0F 7A 32 62 16 FE :0B
24B0 C4 30 8C 02 20 07 48 45 58 32 24 28 00 34 10 17 :67
24C0 0F 10 32 62 10 27 00 10 17 10 52 CC 4E C5 34 06 :8C
24D0 17 0F 53 32 62 16 FE 9D 30 8C 02 20 06 53 50 43 :88
24E0 24 28 00 34 10 17 0E EA 32 62 10 27 00 10 17 10 :A1
24F0 2C CC 4E B6 34 06 17 0F 2D 32 62 16 FE 77 9E 16 :5C
---------------------------------------------------------
Sum: F4 1C AF 31 01 3E D0 5E E4 7B E1 4F FC 0E D7 B8 :85
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2500 4F E6 84 36 06 CC 00 3B BD 4D B9 36 06 9E 16 4F :FE
2510 E6 84 36 06 DC 0E BD 4D B9 AA C0 EA C0 10 27 00 :9E
2520 03 16 FD 68 17 10 12 CC 4E B0 34 06 17 0E F7 32 :09
2530 62 16 FE 41 30 8C 02 20 06 43 4F 4E 53 54 00 34 :56
2540 10 17 0E 8E 32 62 10 27 00 08 CC 03 00 DD 20 16 :78
2550 00 19 30 8C 02 20 04 44 45 46 00 34 10 17 0E 72 :A5
2560 32 62 10 27 00 9B CC 02 00 DD 20 DC 16 83 00 01 :A7
2570 DD 16 DC 16 C3 00 01 DD 16 17 0D 17 DC 22 83 00 :58
2580 03 10 27 00 1B BD 4D 17 BD 4D 0B 44 4F 55 42 4C :01
2590 45 20 44 45 46 20 45 52 52 4F 52 2E 00 16 1A A8 :E4
25A0 9E 16 4F E6 84 83 00 3D 10 27 00 03 16 0C CD DC :32
25B0 08 83 00 02 DD 08 DC 18 9E 08 ED 84 DC 16 C3 00 :32
25C0 01 DD 16 17 0F 73 9E 08 EC 84 DD 18 DC 08 C3 00 :3F
25D0 02 DD 08 DC 1A 83 00 01 10 27 00 03 16 0C 9D DC :36
25E0 20 9E 18 ED 84 DC 24 9E 18 ED 02 DC 04 83 00 03 :52
25F0 DD 04 9E 16 4F E6 84 83 00 2C 10 27 FF 74 16 FC :B9
---------------------------------------------------------
Sum: A7 63 6D 5F DE B3 66 A6 F6 BB 2E B5 68 41 47 E9 :E0
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2600 8B 30 8C 02 20 06 47 4F 54 4F 20 00 34 10 17 0D :30
2610 C1 32 62 10 27 00 17 17 0C 79 CC 00 16 9E 04 E7 :AA
2620 84 DC 04 C3 00 03 DD 04 17 0E AE 16 FC 5E 30 8C :0A
2630 02 20 07 47 4F 53 55 42 20 00 34 10 17 0D 93 32 :F6
2640 62 10 27 02 00 17 0C 4B 17 00 03 16 FC 3E DC 08 :57
2650 83 00 04 DD 08 DC 18 9E 08 ED 84 DC 22 9E 08 ED :08
2660 02 CC 00 00 DD 26 9E 16 4F E6 84 83 00 5B 10 26 :52
2670 01 48 9E 16 4F E6 01 83 00 5D 10 26 00 0A DC 16 :45
2680 C3 00 02 DD 16 16 01 32 DC 16 C3 00 01 DD 16 DC :86
2690 26 C3 00 01 DD 26 9E 16 4F E6 84 83 00 22 10 26 :35
26A0 00 A5 CC 30 8C 9E 04 ED 84 CC 02 20 9E 04 ED 02 :BF
26B0 DC 04 C3 00 05 DD 04 DC 04 DD 28 CC 00 00 DD 2A :41
26C0 DC 16 C3 00 01 DD 16 DC 2A C3 00 01 DD 2A 9E 16 :2E
26D0 4F E6 84 9E 04 E7 84 DC 04 C3 00 01 DD 04 9E 16 :FF
26E0 4F E6 84 36 06 CC 00 22 BD 4D B9 36 06 9E 16 4F :E5
```

拡張Kコンパイラ　ダンプ・リスト

```
26F0 E6 84 36 06 DC 0E BD 4D B9 AA C0 EA C0 27 C1 DC :2B
-----------------------------------------------------
Sum: DF 54 54 F9 35 B0 51 66 5C 28 D3 52 9A 50 B1 68 :C8

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2700 2A 36 06 CC 00 7F BD 4D A9 36 06 9E 16 4F E6 84 :0D
2710 36 06 DC 0E BD 4D B9 AA C0 EA C0 10 27 00 03 16 :4D
2720 19 7E CC 00 00 9E 04 E7 1F DC 2A 9E 28 E7 1F CC :A9
2730 36 10 9E 04 ED 84 DC 16 C3 00 01 DD 16 DC 04 C3 :A5
2740 00 02 DD 04 16 00 11 17 0D EF CC 36 06 9E 04 ED :B4
2750 84 DC 04 C3 00 02 DD 04 9E 16 4F E6 84 83 00 2C :26
2760 10 27 FF 24 9E 16 4F E6 84 83 00 5D 10 27 00 03 :E1
2770 16 19 2D DC 16 C3 00 01 DD 16 DC 26 83 00 00 10 :9A
2780 2F 00 37 CC 00 34 9E 04 E7 1E CC 00 02 DD 1C DC :B0
2790 26 36 06 DC 1C 16 00 1C CC 37 10 9E 04 ED 84 CC :7E
27A0 34 10 9E 04 ED 02 DC 04 C3 00 04 DD 04 DC 1C C3 :18
27B0 00 01 DD 1C A3 C4 2F E0 33 42 CC 00 17 9E 04 E7 :51
27C0 84 DC 04 C3 00 03 DD 04 9E 08 EC 84 DD 18 9E 08 :BC
27D0 EC 02 DD 22 DC 08 C3 00 04 DD 08 9E 18 4F E6 84 :EC
27E0 83 00 02 10 27 00 06 17 0C EF 16 00 0F CC 00 BD :82
27F0 9E 04 E7 1D 9E 18 EC 02 9E 04 ED 1E DC 26 83 00 :7C
-----------------------------------------------------
Sum: 73 11 DB 7F C1 FC CE 17 4C 09 8B 83 99 F7 D7 F0 :3A

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2800 00 10 2F 00 3F CC 00 32 9E 04 E7 84 DC 04 C3 00 :2C
2810 02 DD 04 DC 26 83 00 08 10 2C 00 12 CC 00 60 36 :20
2820 06 DC 26 58 49 E3 C1 9E 04 E7 1F 16 00 16 CC 00 :ED
2830 E9 9E 04 E7 1F DC 26 58 49 9E 04 ED 84 DC 04 C3 :EA
2840 00 02 DD 04 39 30 8C 02 20 07 52 45 54 55 52 4E :E1
2850 00 34 10 17 0B 7C 32 62 10 27 00 11 CC 00 39 9E :61
2860 04 E7 84 DC 04 C3 00 01 DD 04 16 FA 1F 30 8C 02 :E1
2870 20 07 52 45 50 45 41 54 00 34 10 17 0B 54 32 62 :36
2880 10 27 00 1D DC 08 83 00 0A DD 08 DC 12 9E 08 ED :2B
2890 08 CC FF FD 9E 08 ED 84 DC 04 9E 08 ED 02 16 F9 :6B
28A0 EB 30 8C 02 20 07 55 4E 54 49 4C 20 00 34 10 17 :D7
28B0 0B 20 32 62 10 27 00 78 9E 08 EC 84 83 FF FD 10 :13
28C0 27 00 1F BD 4D 17 BD 4D 0B 55 4E 54 49 4C 20 57 :7F
28D0 49 54 48 4F 55 54 20 52 45 50 45 41 54 2E 00 16 :02
28E0 F8 33 17 0C 54 17 17 F0 9E 08 EC 02 93 04 83 FF :6D
28F0 81 10 2F 00 1A DC 2C 9E 04 E7 84 DC 04 C3 00 02 :94
-----------------------------------------------------
Sum: 0C 65 8A ED 1F 5E CB 60 D2 E1 63 FB 2C E3 0A C4 :7E

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2900 DD 04 9E 08 EC 02 93 04 9E 04 E7 1F 16 00 17 DC :BD
2910 2C 9E 04 ED 84 DC 04 C3 00 04 DD 04 9E 08 EC 02 :5B
2920 93 04 9E 04 ED 1E DC 08 C3 00 0A DD 08 16 F9 5C :45
2930 30 8C 02 20 07 57 48 49 4C 45 20 00 34 10 17 0A :E3
2940 91 32 62 10 27 00 36 DC 08 83 00 0A DD 08 DC 12 :D6
2950 9E 08 ED 08 CC FF FC 9E 08 ED 84 DC 04 9E 08 ED :EC
2960 02 17 0B D5 17 17 71 DC 2C 9E 04 ED 84 DC 04 C3 :56
2970 00 04 DD 04 DC 04 9E 08 ED 04 16 F9 0F 30 8C 02 :38
2980 20 05 57 45 4E 44 00 34 10 17 0A 46 32 62 10 27 :C9
2990 00 58 9E 08 EC 84 83 FF FC 10 27 00 1D BD 4D 17 :61
29A0 BD 4D 0B 57 45 4E 44 20 57 49 54 48 4F 55 54 20 :B7
29B0 57 48 49 4C 45 2E 00 16 F7 5B CC 00 16 9E 04 E7 :7A
29C0 84 DC 04 C3 00 03 DD 04 9E 08 EC 02 93 04 9E 04 :D8
29D0 ED 1E 9E 08 EC 04 DD 2E DC 04 93 2E 9E 2E ED 1E :24
29E0 DC 08 C3 00 0A DD 08 16 F8 A2 30 8C 02 20 04 49 :71
29F0 46 20 00 34 10 17 09 DA 32 62 10 27 00 48 17 0B :D9
-----------------------------------------------------
Sum: C4 9B 27 F9 14 AC 8E 01 D4 3A 9C 3D 4B 8C E2 C3 :31

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2A00 38 17 16 D4 30 8C 02 20 05 54 48 45 4E 00 34 10 :8F
2A10 17 09 BF 32 62 DC 08 83 00 0A DD 08 DC 12 9E 08 :5D
2A20 ED 08 CC FF FE 9E 08 ED 84 DC 2C 9E 04 ED 84 DC :CC
2A30 04 C3 00 04 DD 04 DC 04 9E 08 ED 02 CC 00 00 9E :8B
2A40 08 ED 04 16 F8 46 30 8C 02 20 05 45 4C 53 45 00 :59
2A50 34 10 17 09 7D 32 62 10 27 00 4A 9E 08 EC 84 83 :8F
2A60 FF FE 10 27 00 1A BD 4D 17 BD 4D 0B 45 4C 53 45 :AD
2A70 20 57 49 54 48 4F 55 54 20 49 46 2E 00 16 F6 95 :D2
2A80 9E 08 EC 02 DD 2E CC 00 16 9E 04 E7 84 DC 04 C3 :31
2A90 00 03 DD 04 DC 04 9E 08 ED 02 DC 04 93 2E 9E 2E :C6
2AA0 ED 1E 16 F7 E7 30 8C 02 20 03 46 49 00 34 10 17 :CA
2AB0 09 20 32 62 10 27 00 3B 9E 08 EC 84 83 FF FE 10 :D5
2AC0 27 00 18 BD 4D 17 BD 4D 0B 46 49 20 57 49 54 48 :60
2AD0 4F 55 54 20 49 46 2E 00 16 F6 3A 9E 08 EC 02 DD :8C
2AE0 2E DC 04 93 2E 9E 2E ED 1E DC 08 C3 00 0A DD 08 :3C
2AF0 16 F7 99 30 8C 02 20 05 46 4F 52 20 00 34 10 17 :EB
-----------------------------------------------------
Sum: E9 AE 2F A2 2A 71 C1 55 CD 7A 0F 62 8C 50 5B 4B :53

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2B00 08 D0 32 62 10 27 00 F4 17 07 88 9E 16 4F E6 84 :AA
2B10 83 00 3D 10 27 00 03 16 07 62 DC 22 83 00 03 10 :0D
2B20 26 00 20 CC 01 00 9E 18 ED 84 DC 0C 9E 18 ED 02 :C7
2B30 DC 0C C3 00 02 DD 0C DC 0C 83 01 02 10 26 00 03 :3D
2B40 16 15 78 9E 18 EC 02 DD 30 DC 16 C3 00 01 DD 16 :FD
2B50 17 09 E6 30 8C 02 20 03 54 4F 00 34 10 17 08 72 :5F
2B60 32 62 83 00 00 10 26 00 03 16 07 10 CC 00 DD 9E :C4
2B70 04 E7 84 DC 30 9E 04 E7 01 DC 04 C3 00 02 DD 04 :8B
2B80 DC 08 83 00 0A DD 08 DC 12 9E 08 ED 08 CC FF FF :A9
2B90 9E 08 ED 84 DC 30 9E 08 ED 02 17 09 9C CC 36 06 :7C
2BA0 9E 04 ED 84 CC 00 DC 9E 04 E7 02 DC 30 9E 04 E7 :DB
2BB0 03 CC 00 16 9E 04 E7 04 DC 04 C3 00 07 DD 04 DC :D9
2BC0 04 9E 08 ED 04 30 8C 02 20 05 53 54 45 50 00 34 :EE
2BD0 10 17 07 FE 32 62 10 27 00 18 17 09 5C DC 1A 83 :04
2BE0 00 01 10 27 00 03 16 06 93 DC 24 9E 08 ED 06 16 :99
2BF0 00 07 CC 00 01 9E 08 ED 06 16 F6 90 30 8C 02 20 :E7
-----------------------------------------------------
Sum: 1F E0 FF 18 95 E4 1C 67 37 27 CA F5 D7 5F D4 78 :B1

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2C00 05 4E 45 58 54 00 34 10 17 07 C7 32 62 10 27 00 :38
2C10 ED 9E 08 EC 84 83 FF FF 10 27 00 1B BD 4D 17 BD :B4
2C20 4D 0B 4E 45 58 54 20 57 49 54 48 4F 55 54 20 46 :51
2C30 4F 52 2E 00 16 F4 DE CC 00 DC 9E 04 E7 84 9E 08 :12
2C40 EC 02 9E 04 E7 01 CC 00 C3 9E 04 E7 02 CC 00 DD :3B
2C50 9E 04 E7 05 9E 08 EC 02 9E 04 E7 06 CC 00 A3 9E :BE
2C60 04 E7 07 CC 00 C4 9E 04 E7 08 DC 04 C3 00 09 DD :9C
2C70 04 9E 08 EC 06 9E 04 ED 1A 9E 08 EC 06 83 00 00 :60
2C80 10 2F 00 08 CC 10 2F DD 32 16 00 05 CC 10 2C DD :61
2C90 32 9E 08 EC 04 DD 2E DC 04 83 00 02 93 2E 9E 2E :C5
2CA0 ED 1E 9E 08 EC 04 93 04 83 FF 81 10 2F 00 24 DC :7A
2CB0 32 9E 04 E7 84 9E 08 EC 04 93 04 83 00 02 9E 04 :93
2CC0 E7 01 CC 33 42 9E 04 ED 02 DC 04 C3 00 04 DD 04 :42
2CD0 16 00 21 DC 32 9E 04 ED 84 9E 08 EC 04 93 04 83 :08
2CE0 00 04 9E 04 ED 02 CC 33 42 9E 04 ED 04 DC 04 C3 :0C
2CF0 00 06 DD 04 DC 08 C3 00 0A DD 08 16 F5 8E 30 8C :D2
-----------------------------------------------------
Sum: 7E 68 6F 44 4E 0B 1A DB 61 C6 19 C9 7D C5 49 24 :9F

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2D00 02 20 05 50 4F 4B 45 00 34 10 17 06 C5 32 62 10 :20
2D10 27 00 BB 17 08 23 9E 16 4F E6 84 83 00 2C 10 27 :77
2D20 00 03 16 05 57 DC 16 C3 00 01 DD 16 DC 1A 83 00 :97
2D30 01 10 26 00 0A DC 04 83 00 03 DD 04 16 00 21 DC :9B
2D40 1A 83 00 02 10 26 00 0A CC 00 9E 9E 04 E7 1E 16 :06
2D50 00 0E CC 36 06 9E 04 ED 84 DC 04 C3 00 02 DD 04 :AF
2D60 DC 08 83 00 04 DD 08 DC 1A 9E 08 ED 84 DC 24 9E :FB
2D70 08 ED 02 17 07 C3 9E 08 EC 84 83 00 01 10 26 00 :A8
2D80 19 CC 00 F7 9E 04 E7 84 DC 04 C3 00 03 DD 04 9E :0E
2D90 08 EC 02 9E 04 ED 1E 16 00 2A 9E 08 EC 84 83 00 :7C
2DA0 02 10 26 00 11 CC E7 84 9E 04 ED 84 DC 04 C3 00 :36
2DB0 02 DD 04 16 00 0E CC E7 D1 9E 04 ED 84 DC 04 C3 :41
2DC0 00 02 DD 04 DC 08 C3 00 04 DD 08 16 F4 BE 30 8C :F7
2DD0 02 20 06 43 50 4F 4B 45 00 34 10 17 05 F4 32 62 :82
2DE0 10 27 00 76 17 0B 7F DC 1A 83 00 02 10 27 00 03 :03
2DF0 16 04 89 DC 24 DD 34 CC 00 9E 9E 04 E7 1E 17 06 :E2
-----------------------------------------------------
Sum: 75 AB E5 FF F3 94 20 29 42 FA 8A 9D 7F 85 22 23 :80

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2E00 C1 DC 16 C3 00 01 DD 16 17 07 2E 9E 16 4F E6 84 :23
2E10 83 00 23 10 26 00 14 CC ED 81 9E 04 ED 84 DC 16 :2F
2E20 C3 00 01 DD 16 17 06 9A 16 00 07 CC E7 80 9E 04 :60
2E30 ED 84 DC 04 C3 00 02 DD 04 9E 16 4F E6 84 83 00 :E7
2E40 2C 27 BE CC 00 9F 9E 04 E7 84 DC 34 9E 04 E7 01 :23
2E50 DC 04 C3 00 02 DD 04 16 F4 32 30 8C 02 20 05 43 :E8
2E60 41 4C 4C 00 34 10 17 05 69 32 62 10 27 00 2E 17 :B2
2E70 06 C7 DC 1A 83 00 01 10 26 00 0A CC 00 BD 9E 04 :B2
2E80 E7 1D 16 F4 07 CC 1F 01 9E 04 ED 84 CC AD 84 9E :AF
2E90 04 ED 02 DC 04 C3 00 04 DD 04 16 F3 EF 30 8C 02 :31
2EA0 20 05 43 4F 44 45 00 34 10 17 05 26 32 62 10 27 :91
2EB0 00 60 DC 16 83 00 01 DD 16 DC 16 C3 00 01 DD 16 :72
2EC0 17 05 FF 17 0A A0 DC 1A 83 00 01 10 27 00 03 16 :A6
2ED0 03 AA 9E 16 4F E6 84 83 00 23 10 26 00 17 DC 04 :ED
2EE0 83 00 01 DD 04 DC 24 9E 04 ED 16 DC 16 C3 00 01 :C8
2EF0 DD 16 16 00 0D DC 04 83 00 02 DD 04 DC 24 9E 04 :FE
-----------------------------------------------------
Sum: C8 D2 AA D9 F4 B6 5B 5C B0 1B 8B CF 9D F6 15 F9 :44

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2F00 E7 1F 17 05 BD 9E 16 4F E6 84 83 00 2C 27 AA 16 :E2
2F10 F3 7A 30 8C 02 20 04 45 4E 44 00 34 10 17 04 B2 :37
2F20 32 62 10 27 00 26 CC C6 00 9E 04 ED 84 CC 1F 9B :1C
2F30 9E 04 ED 02 CC 7E 94 9E 04 ED 04 CC 00 8D 9E 04 :FD
2F40 E7 06 DC 04 C3 00 07 DD 04 16 F3 40 9E 16 4F E6 :AA
2F50 84 34 06 17 05 02 32 62 10 27 02 9C 17 03 34 9E :31
2F60 16 4F E6 84 83 00 5B 10 26 00 06 17 F6 E0 16 F3 :DF
2F70 1B 9E 16 4F E6 84 83 00 3B 10 26 00 6F DC 22 83 :6C
2F80 00 03 10 26 00 10 CC 00 00 9E 18 ED 84 DC 04 9E :BA
2F90 18 ED 02 16 F2 F6 9E 18 4F E6 01 83 00 00 10 26 :AA
2FA0 00 1B BD 4D 17 BD 4D 0B 44 4F 55 42 4C 45 20 44 :70
2FB0 45 46 20 45 52 52 4F 52 2E 00 16 10 8B 9E 18 EC :B6
2FC0 02 DD 36 CC 00 00 9E 18 E7 01 9E 36 EC 1E DD 2E :68
2FD0 DC 04 93 36 9E 36 ED 1E DC 2E DD 36 DC 36 83 00 :3A
```

```
2FE0 00 26 E7 DC 04 9E 18 ED 02 16 F2 A0 DC 22 83 00 :BB
2FF0 03 10 26 00 20 CC 01 00 9E 18 ED 84 DC 0C 9E 18 :EB
---------------------------------------------------------
Sum: 84 8E E7 54 D9 9D 3B DF D1 D0 8A 32 B5 AD F3 9B :2A

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3000 ED 02 DC 0C C3 00 02 DD 0C DC 0C 83 01 02 10 26 :29
3010 00 03 16 10 A6 9E 18 EC 02 DD 38 9E 16 4F E6 84 :F5
3020 83 00 3A 10 26 00 58 DC 16 C3 00 01 DD 16 17 05 :10
3030 08 DC 1A 83 00 01 10 26 00 08 CC 00 E7 DD 3A 16 :A0
3040 00 AE CC 00 9E 9E 04 E7 84 DC 38 9E 04 E7 01 CC :8F
3050 30 8B 9E 04 ED 02 CC 36 10 9E 04 ED 04 DC 04 C3 :94
3060 00 06 DD 04 DC 16 C3 00 02 DD 16 17 04 CB CC E7 :2A
3070 D1 9E 04 ED 84 DC 04 C3 00 02 DD 04 16 F2 0D 9E :1D
3080 16 4F E6 84 83 00 28 10 26 01 40 DC 16 C3 00 01 :A7
3090 DD 16 17 04 A4 DC 1A 83 00 01 10 26 00 0E CC 00 :3C
30A0 ED DD 3A DC 24 58 49 DD 24 16 00 44 CC 58 49 9E :0B
30B0 04 ED 84 CC 00 9E 9E 04 E7 02 DC 38 9E 04 E7 03 :0A
30C0 CC 30 8B 9E 04 ED 04 CC 36 10 9E 04 ED 06 DC 04 :A1
30D0 C3 00 08 DD 04 DC 16 C3 00 02 DD 16 17 04 5A CC :97
30E0 ED D1 9E 04 ED 84 DC 04 C3 00 02 DD 04 16 F1 9C :FA
30F0 DC 04 83 00 03 DD 04 DC 08 83 00 02 DD 08 DC 24 :95
---------------------------------------------------------
Sum: B5 F2 00 53 BD 2D 3C 8E EC 8C E8 3F 62 19 24 0B :F7

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3100 9E 08 ED 84 DC 16 C3 00 02 DD 16 17 04 2B CC 00 :D3
3110 9E 9E 04 E7 84 DC 38 9E 04 E7 01 DC 3A 9E 04 E7 :E8
3120 02 9E 08 EC 84 DD 24 DC 08 C3 00 02 DD 08 DC 24 :A7
3130 83 00 00 10 26 00 11 CC 00 84 9E 04 E7 03 DC 04 :86
3140 C3 00 04 DD 04 16 F1 44 DC 24 36 06 CC 00 10 BD :C8
3150 4D A1 36 06 DC 24 36 06 CC FF EF BD 4D A9 A4 C0 :37
3160 E4 C0 10 27 00 19 DC 24 36 06 CC 00 1F A4 C0 E4 :63
3170 C0 9E 04 E7 03 DC 04 C3 00 04 DD 04 16 F1 0D DC :C4
3180 24 36 06 CC 00 80 BD 4D A1 36 06 DC 24 36 06 CC :9B
3190 FF 7F BD 4D A9 A4 C0 E4 C0 10 27 00 17 CC 00 88 :DB
31A0 9E 04 E7 03 DC 24 9E 04 E7 04 DC 04 C3 00 05 DD :9E
31B0 04 16 F0 D8 CC 00 89 9E 04 E7 03 DC 24 9E 04 ED :52
31C0 04 DC 04 C3 00 06 DD 04 16 F0 C1 9E 16 4F E6 84 :C2
31D0 83 00 3D 10 26 00 21 DC 16 C3 00 01 DD 16 17 03 :DA
31E0 58 CC 00 DD 9E 04 E7 84 DC 38 9E 04 E7 01 DC 04 :8C
31F0 C3 00 02 DD 04 16 F0 94 9E 16 4F E6 84 83 00 25 :55
---------------------------------------------------------
Sum: DC BA 24 D9 06 66 B0 42 DE 6A 3D 05 D0 9B F1 1A :F1

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3200 10 26 00 78 DC 16 C3 00 01 DD 16 17 07 58 9E 16 :81
3210 4F E6 84 36 06 CC 00 3D BD 4D C1 36 06 DC 1A 36 :31
3220 06 CC 00 01 BD 4D C1 AA C0 EA C0 10 27 00 03 16 :02
3230 00 4A DC 04 83 00 03 DD 04 DC 24 58 49 DD 3C DC :27
3240 16 C3 00 01 DD 16 17 02 F0 DC 3C 83 00 10 10 2C :BD
3250 00 13 CC ED 60 D3 3C 9E 04 ED 84 DC 04 C3 00 02 :F3
3260 DD 04 16 00 14 CC ED E8 9E 04 ED 84 DC 3C 9E 04 :79
3270 E7 02 DC 04 C3 00 03 DD 04 16 F0 10 BD 4D 17 BD :64
3280 4D 0B 53 59 4E 54 41 58 20 45 52 52 4F 52 2E 00 :17
3290 16 0D B5 17 02 2C 9E 16 4F E6 84 34 06 17 01 BB :94
32A0 32 62 83 00 00 12 12 12 10 26 00 03 16 FF CD CC :34
32B0 00 00 DD 1C CC 00 07 36 06 DC 1C 16 00 13 DC 1C :21
32C0 10 9E 0A 31 AB CC 00 20 E7 A4 DC 1C C3 00 01 DD :A4
32D0 1C A3 C4 2F E9 33 42 CC 00 00 DD 22 9E 16 4F E6 :C4
32E0 84 34 06 17 01 72 32 62 36 06 9E 16 4F E6 84 34 :B9
32F0 06 17 01 47 32 62 AA C0 EA C0 10 27 00 2F DC 22 :71
---------------------------------------------------------
Sum: 8A 04 5B EF 19 49 E0 ED A4 6A B1 C2 35 13 44 E6 :FA

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3300 83 00 08 10 2C 00 1C DC 22 10 9E 0A 31 AB 9E 16 :29
3310 4F E6 84 34 06 17 01 7D 32 62 E7 A4 DC 22 C3 00 :68
3320 01 DD 22 DC 16 C3 00 01 DD 16 16 FF AF CC 00 00 :39
3330 DD 22 DC 06 DD 18 9E 18 4F E6 84 83 00 FF 10 26 :FD
3340 00 48 CC 00 00 DD 1C CC 00 03 36 06 DC 1C 16 00 :26
3350 1D DC 1C C3 00 02 58 49 10 9E 18 31 AB DC 1C 9E :B3
3360 0A 58 49 EC 8B ED A4 DC 1C C3 00 01 DD 1C A3 C4 :CF
3370 2F DF 33 42 CC 00 00 9E 18 E7 84 CC 00 FF 9E 18 :F1
3380 E7 0C CC 00 03 DD 22 16 00 3C CC 00 00 DD 1C DC :B4
3390 1C C3 00 04 9E 18 E6 8B 4F 36 06 DC 1C 9E 0A E6 :1B
33A0 8B 4F BD 4D C1 10 27 00 0A DC 18 C3 00 0C DD 18 :9E
33B0 16 00 13 DC 1C C3 00 01 DD 1C DC 1C 83 00 08 26 :87
33C0 CE CC 00 01 DD 22 DC 22 83 00 00 10 2F FF 67 16 :D6
33D0 00 F0 EC 62 DD 2E CC 00 00 DD 1C DC 1C 9E 2E E6 :B8
33E0 8B 4F 83 00 00 10 27 00 2E DC 1C 9E 2E E6 8B 4F :46
33F0 36 06 DC 1C 9E 16 E6 8B 4F 34 06 17 00 97 32 62 :24
---------------------------------------------------------
Sum: 39 6F D5 C3 52 FC B7 50 FA 10 F5 90 38 4C 41 63 :4C

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3400 BD 4D C1 10 27 00 06 CC 00 00 DD 2E 39 DC 1C C3 :D3
3410 00 01 DD 1C 16 FF C4 DC 16 D3 1C DD 16 17 00 A2 :60
3420 CC 00 01 DD 2E 39 CC 00 BD 9E 04 E7 84 DC 04 C3 :4A
3430 00 03 DD 04 EC 62 9E 04 ED 1E 39 EC 62 36 06 CC :6E
3440 00 30 BD 4D 99 36 06 EC 62 36 06 CC 00 39 BD 4D :A8
3450 B1 A4 C0 E4 C0 DD 2E 39 EC 62 36 06 CC 00 41 BD :51
3460 4D 99 36 06 EC 62 36 06 CC 00 5A BD 4D B1 A4 C0 :F1
3470 E4 C0 36 06 EC 62 36 06 CC 00 61 BD 4D 99 36 06 :76
3480 EC 62 36 06 CC 00 7A BD 4D B1 A4 C0 E4 C0 AA C0 :FD
3490 EA C0 DD 2E 39 EC 62 36 06 CC 00 60 BD 4D A9 36 :8D
34A0 06 EC 62 36 06 CC 00 7B BD 4D A1 A4 C0 E4 C0 10 :9A
34B0 27 00 0A EC 62 83 00 20 DD 40 16 00 04 EC 62 DD :84
34C0 40 39 9E 16 4F E6 84 83 00 20 10 26 00 0A DC 16 :BB
34D0 C3 00 01 DD 16 16 FF EA 39 DC 22 83 00 03 10 26 :A9
34E0 00 15 CC 00 00 9E 04 ED 1E CC 00 01 9E 18 ED 84 :82
34F0 DC 04 9E 18 ED 02 39 9E 18 4F E6 01 83 00 00 10 :3D
---------------------------------------------------------
Sum: 4D DE ED AB 47 48 70 63 02 48 A0 99 21 8A 4C 77 :16

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3500 26 00 0B 9E 18 EC 02 93 04 9E 04 ED 1E 39 9E 18 :08
3510 EC 02 9E 04 ED 1E DC 04 9E 18 ED 02 39 17 00 19 :89
3520 9E 16 4F E6 84 83 00 29 10 27 00 03 16 0A FF DC :4E
3530 16 C3 00 01 DD 16 16 FF 89 17 FF 86 17 00 A3 9E :5F
3540 16 4F E6 84 DD 36 30 8C 02 20 04 41 4E 44 00 34 :CB
3550 10 17 FE 7E 32 62 36 06 30 8C 02 20 03 4F 52 00 :F5
3560 34 10 17 FE 6D 32 62 AA C0 EA C0 10 27 00 70 DC :F1
3570 08 83 00 02 DD 08 DC 36 9E 08 ED 84 CC 36 06 9E :41
3580 04 ED 84 DC 04 C3 00 02 DD 04 17 00 55 9E 08 EC :F9
3590 84 DD 2E DC 08 C3 00 02 DD 08 DC 2E 83 00 41 10 :FB
35A0 26 00 08 CC 00 84 DD 2E 16 00 05 CC 00 8A DD 2E :05
35B0 DC 2E C3 00 20 9E 04 E7 84 CC 00 C0 9E 04 E7 01 :10
35C0 DC 2E C3 00 60 9E 04 E7 02 CC 00 C0 9E 04 E7 03 :D0
35D0 DC 04 C3 00 04 DD 04 CC 00 05 DD 1A 16 FF 67 16 :E2
35E0 FE E0 17 FE DD 17 00 E7 30 8C 02 20 03 3C 3E 00 :29
35F0 34 10 17 FD DD 32 62 10 27 00 08 CC 4D C1 DD 2C :EB
---------------------------------------------------------
Sum: 9C EE 24 0A 09 E1 E3 F4 78 C7 82 ED 42 4F 7E C9 :FF

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3600 16 00 81 30 8C 02 20 03 3C 3D 00 34 10 17 FD C2 :0B
3610 32 62 10 27 00 08 CC 4D B1 DD 2C 16 00 66 30 8C :DE
3620 02 20 03 3E 3D 00 34 10 17 FD A7 32 62 10 27 00 :6A
3630 08 CC 4D 99 DD 2C 16 00 4B 30 8C 02 20 02 3D 00 :41
3640 34 10 17 FD 8D 32 62 10 27 00 08 CC 4D B9 DD 2C :93
3650 16 00 31 30 8C 02 20 02 3E 00 34 10 17 FD 73 32 :62
3660 62 10 27 00 08 CC 4D A9 DD 2C 16 00 17 30 8C 02 :57
3670 20 02 3C 00 34 10 17 FD 59 32 62 10 27 00 4D CC :F3
3680 4D A1 DD 2C DC 08 83 00 02 DD 08 DC 2C 9E 08 ED :E0
3690 84 CC 36 06 9E 04 ED 84 DC 04 C3 00 02 DD 04 17 :3C
36A0 00 2D 9E 08 EC 84 DD 2C DC 08 C3 00 02 DD 08 CC :A6
36B0 00 BD 9E 04 E7 84 DC 04 C3 00 03 DD 04 DC 2C 9E :F7
36C0 04 ED 1E DC 1A DD 42 CC 00 04 DD 1A 16 FD F3 17 :08
36D0 FD F0 17 01 21 9E 16 4F E6 84 36 06 CC 00 2B BD :83
36E0 4D B9 36 06 9E 16 4F E6 84 36 06 CC 00 2D BD 4D :EE
36F0 B9 AA C0 EA C0 10 27 00 FA DC 08 83 00 02 DD 08 :4C
---------------------------------------------------------
Sum: F6 07 06 66 E1 FB 13 CD CB 28 C5 92 4A D5 B2 11 :51

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3700 9E 16 4F E6 84 9E 08 ED 84 DC 16 C3 00 01 DD 16 :2D
3710 CC 36 06 9E 04 ED 84 DC 04 C3 00 02 DD 04 17 00 :B8
3720 D5 9E 08 EC 84 DD 2E DC 08 C3 00 02 DD 08 DC 2E :8E
3730 83 00 2B 10 26 00 51 DC 1A 83 00 01 10 26 00 17 :FC
3740 DC 04 83 00 02 DD 04 CC 00 C3 9E 04 E7 1D DC 24 :7B
3750 9E 04 ED 1E 16 00 2E DC 1A 83 00 02 10 26 00 17 :B9
3760 DC 04 83 00 02 DD 04 CC 00 D3 9E 04 E7 1E DC 24 :8C
3770 9E 04 E7 1F 16 00 0E CC E3 C1 9E 04 ED 84 DC 04 :2F
3780 C3 00 02 DD 04 16 00 63 DC 1A 83 00 01 10 26 00 :CF
3790 17 DC 04 83 00 02 DD 04 CC 00 83 9E 04 E7 1D DC :2E
37A0 24 9E 04 ED 1E 16 00 43 DC 1A 83 00 02 10 26 00 :DB
37B0 17 DC 04 83 00 02 DD 04 CC 00 93 9E 04 E7 1E DC :3F
37C0 24 9E 04 E7 1F 16 00 23 CC A3 C1 9E 04 ED 84 CC :14
37D0 53 43 9E 04 ED 02 CC C3 00 9E 04 ED 04 CC 00 01 :16
37E0 9E 04 E7 06 DC 04 C3 00 07 DD 04 CC 00 05 DD 1A :E2
37F0 16 FE E2 16 FC CC 17 FC C9 17 01 6A 9E 16 4F E6 :1B
---------------------------------------------------------
Sum: F6 33 DB 94 68 3A AF 51 93 28 D6 D3 46 DA 9B 43 :9C

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3800 84 36 06 CC 00 2A BD 4D B9 36 06 9E 16 4F E6 84 :22
3810 36 06 CC 00 2F BD 4D B9 AA C0 EA C0 10 27 01 43 :89
3820 DC 08 83 00 02 DD 08 9E 16 4F E6 84 9E 08 ED 84 :D2
3830 DC 16 C3 00 01 DD 16 CC 36 06 9E 04 ED 84 DC 04 :A4
3840 C3 00 02 DD 04 17 01 1E 9E 08 EC 84 DD 2E DC 08 :E1
3850 C3 00 02 DD 08 DC 2E 83 00 2A 10 26 00 80 DC 1A :0D
3860 36 06 CC 00 01 BD 4D B9 36 06 DC 24 36 06 CC 00 :10
3870 02 BD 4D B9 36 06 DC 24 36 06 CC 00 04 BD 4D B9 :D0
3880 AA C0 EA C0 36 06 DC 24 36 06 CC 00 08 BD 4D B9 :23
3890 AA C0 EA C0 36 06 DC 24 36 06 CC 00 10 BD 4D B9 :2B
38A0 AA C0 EA C0 A4 C0 E4 C0 10 27 00 25 DC 04 83 00 :DB
38B0 05 DD 04 CC 58 49 9E 04 ED 84 DC 04 C3 00 02 DD :E8
38C0 04 DC 24 47 56 DD 24 DC 24 83 00 01 26 E5 16 00 :47
```

拡張Kコンパイラ ダンプ・リスト

```
38D0 0A CC 4D 1F 34 06 17 FB 4D 32 62 16 00 7D DC 1A :F8
38E0 36 06 CC 00 01 BD 4D B9 36 06 DC 24 36 06 CC 00 :10
38F0 02 BD 4D B9 36 06 DC 24 36 06 CC 00 04 BD 4D B9 :D0
-----------------------------------------------------------
Sum: 79 A5 81 6A 9E 12 1E AE FF 01 96 18 DF 16 AB 4C :1F

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3900 AA C0 EA C0 36 06 DC 24 36 06 CC 00 08 BD 4D B9 :23
3910 AA C0 EA C0 36 06 DC 24 36 06 CC 00 10 BD 4D B9 :2B
3920 AA C0 EA C0 A4 C0 E4 C0 10 27 00 25 DC 04 83 00 :DB
3930 05 DD 04 CC 47 56 9E 04 ED 84 DC 04 C3 00 02 DD :E4
3940 04 DC 24 47 56 DD 24 DC 24 83 00 01 26 E5 16 00 :47
3950 0A CC 4D 3B 34 06 17 FA CD 32 62 CC 00 05 DD 1A :D2
3960 16 FE 99 16 FB 5C 30 8C 02 20 06 49 4E 50 55 54 :8E
3970 00 34 10 17 FA 5C 32 62 10 27 00 12 CC 4D E2 34 :BD
3980 06 17 FA A2 32 62 CC 00 03 DD 1A 16 FB 34 30 8C :14
3990 02 20 04 47 45 54 00 34 10 17 FA 36 32 62 10 27 :5C
39A0 00 12 CC 4D D6 34 06 17 FA 7C 32 62 CC 00 03 DD :08
39B0 1A 16 FB 0E 30 8C 02 20 05 41 42 53 28 00 34 10 :5E
39C0 17 FA 0F 32 62 10 27 00 15 17 FB 51 CC 4D 95 34 :45
39D0 06 17 FA 52 32 62 CC 00 03 DD 1A 16 FA E4 30 8C :73
39E0 02 20 05 4E 4F 54 28 00 34 10 17 F9 E5 32 62 10 :1D
39F0 27 00 15 17 FB 27 CC 4D CD 34 06 17 FA 28 32 62 :62
-----------------------------------------------------------
Sum: 8F 87 C4 E8 31 20 92 88 97 9C 96 C9 BD 26 19 C3 :7E

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3A00 CC 00 03 DD 1A 16 FA BA 30 8C 02 20 05 41 44 52 :4A
3A10 28 00 34 10 17 F9 BB 32 62 10 27 00 2C 17 FA FD :3C
3A20 DC 1A 83 00 02 10 27 00 03 16 F8 50 CC 00 03 DD :BF
3A30 1A CC 00 C6 9E 04 E7 1E CC 1F B8 9E 04 ED 84 DC :E5
3A40 04 C3 00 02 DD 04 16 FA 79 30 8C 02 20 06 48 49 :A8
3A50 47 48 28 00 34 10 17 F9 79 32 62 10 27 00 20 17 :86
3A60 FA BB CC 1F 89 9E 04 ED 84 CC 00 4F 9E 04 E7 02 :E2
3A70 DC 04 C3 00 03 DD 04 CC 00 03 DD 1A 16 FA 43 30 :D0
3A80 8C 02 20 05 4C 4F 57 28 00 34 10 17 F9 44 32 62 :F9
3A90 10 27 00 19 17 FA 86 CC 00 4F 9E 04 E7 84 DC 04 :EF
3AA0 C3 00 01 DD 04 CC 00 03 DD 1A 16 FA 15 30 8C 02 :4E
3AB0 20 05 53 45 58 28 00 34 10 17 F9 16 32 62 10 27 :72
3AC0 00 19 17 FA 58 CC 00 1D 9E 04 E7 84 DC 04 C3 00 :1B
3AD0 01 DD 04 CC 00 03 DD 1A 16 F9 E7 30 8C 02 20 05 :81
3AE0 53 47 4E 28 00 34 10 17 F8 E8 32 62 10 27 00 4A :60
3AF0 17 FA 2A CC 00 03 DD 1A CC 83 00 9E 04 ED 84 CC :2F
-----------------------------------------------------------
Sum: F5 15 78 CE 85 F5 9F 49 3C 1E 61 68 9F BD 68 44 :DD

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3B00 00 27 9E 04 ED 02 CC 0A 2A 9E 04 ED 04 CC 05 CC :E8
3B10 9E 04 ED 06 CC FF FF 9E 04 ED 08 CC 20 03 9E 04 :87
3B20 ED 0A CC CC 00 9E 04 ED 0C CC 00 01 9E 04 E7 0E :8E
3B30 DC 04 C3 00 0F DD 04 16 F9 88 9E 16 4F E6 84 83 :1A
3B40 00 2D 10 26 00 49 DC 16 C3 00 01 DD 16 17 FE 16 :80
3B50 DC 1A 83 00 01 10 26 00 12 DC 24 53 43 C3 00 01 :1C
3B60 DD 24 DC 24 9E 04 ED 1E 16 00 21 CC 53 43 9E 04 :E9
3B70 ED 84 CC C3 00 9E 04 ED 02 CC 00 01 9E 04 E7 04 :EB
3B80 DC 04 C3 00 05 DD 04 CC 00 03 DD 1A 16 F9 33 30 :C1
3B90 8C 02 20 06 50 45 45 4B 28 00 34 10 17 F8 33 32 :B9
3BA0 62 10 27 00 66 17 F9 75 DC 1A 83 00 01 10 26 00 :34
3BB0 17 DC 04 C3 00 01 DD 04 CC 4F F6 9E 04 ED 1C DC :34
3BC0 24 9E 04 ED 1E 16 00 3D DC 1A 83 00 02 10 26 00 :D5
3BD0 18 CC 00 9E 9E 04 E7 1E CC EC 84 9E 04 ED 84 DC :54
3BE0 04 C3 00 02 DD 04 16 00 1C CC 1F 01 9E 04 ED 84 :DB
3BF0 CC 4F E6 9E 04 ED 02 CC 00 84 9E 04 E7 04 DC 04 :4F
-----------------------------------------------------------
Sum: FA 96 4D D7 BF BC E4 83 B4 49 3E 38 18 CD AC 22 :BC

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3C00 C3 00 05 DD 04 CC 00 03 DD 1A 39 9E 16 4F E6 84 :15
3C10 83 00 25 10 26 00 5B DC 16 C3 00 01 DD 16 17 FD :F6
3C20 45 DC 1A 83 00 01 10 27 00 03 16 F6 4F DC 04 83 :B7
3C30 00 01 DD 04 CC 00 EC 9E 04 E7 1E DC 24 58 49 DD :BF
3C40 24 DC 24 83 00 10 10 2C 00 0C CC 00 60 D3 24 9E :C0
3C50 04 E7 1F 16 00 14 CC 00 E8 9E 04 E7 1F DC 24 9E :2E
3C60 04 E7 84 DC 04 C3 00 01 DD 04 CC 00 06 DD 1A 16 :D3
3C70 F8 50 30 8C 02 20 05 52 4E 44 28 00 34 10 17 F7 :89
3C80 51 32 62 10 27 00 15 17 F8 93 CC 4F 09 34 06 17 :48
3C90 F7 94 32 62 CC 00 03 DD 1A 16 F8 26 9E 16 4F E6 :02
3CA0 84 83 00 28 10 26 00 23 DC 16 C3 00 01 DD 16 17 :48
3CB0 F8 87 9E 16 4F E6 84 83 00 29 10 27 00 03 16 03 :EB
3CC0 6D DC 16 C3 00 01 DD 16 16 F7 F7 9E 16 4F E6 84 :87
3CD0 83 00 24 10 26 00 78 CC 00 00 DD 2E CC 00 00 DD :D5
3CE0 24 DC 2E 83 00 09 10 2F 00 07 DC 2E 83 00 07 DD :71
3CF0 2E DC 24 58 49 58 49 58 49 58 49 D3 2E DD 24 DC :90
-----------------------------------------------------------
Sum: B5 3B D6 D3 BD 42 82 26 57 F7 C1 C1 5A 8B 55 5B :A5

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3D00 16 C3 00 01 DD 16 9E 16 4F E6 84 83 00 30 DD 2E :F8
3D10 DC 2E 36 06 CC 00 09 BD 4D A9 36 06 DC 2E 36 06 :50
3D20 CC 00 11 BD 4D A1 A4 C0 E4 C0 36 06 DC 2E 36 06 :12
3D30 CC 00 00 BD 4D A1 AA C0 EA C0 36 06 DC 2E 36 06 :0D
3D40 CC 00 16 BD 4D A9 AA C0 EA C0 27 95 16 00 45 9E :5E
3D50 16 4F E6 84 34 06 17 F6 E2 32 62 10 27 00 51 CC :E0
3D60 00 00 DD 24 DC 24 36 06 CC 00 0A BD 4D 1F 36 06 :78
3D70 9E 16 4F E6 84 E3 C1 83 00 30 DD 24 DC 16 C3 00 :7A
3D80 01 DD 16 9E 16 4F E6 84 34 06 17 F6 AE 32 62 83 :6D
3D90 00 00 26 D0 CC 00 CC 9E 04 E7 84 DC 04 C3 00 03 :41
3DA0 DD 04 DC 24 9E 04 ED 1E CC 00 01 DD 1A 16 F7 12 :71
3DB0 9E 16 4F E6 84 83 00 27 10 26 00 11 9E 16 4F E6 :47
3DC0 01 DD 24 DC 16 C3 00 03 DD 16 16 FF C7 9E 16 4F :8C
3DD0 E6 84 34 06 17 F6 81 32 62 10 27 02 51 17 F4 B3 :0E
3DE0 9E 18 4F E6 84 83 00 03 10 26 00 09 9E 18 EC 02 :D8
3DF0 DD 24 16 FF 9F 9E 16 4F E6 84 83 00 5B 10 26 00 :36
-----------------------------------------------------------
Sum: E8 EA 93 0B 78 BE E3 80 4B 14 F2 E5 75 ED D2 32 :A5

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3E00 0E CC 00 17 DD 2C 17 E8 45 CC 00 06 DD 1A 39 DC :1C
3E10 22 83 00 03 10 26 00 20 CC 01 00 9E 18 ED 84 DC :CE
3E20 0C 9E 18 ED 02 DC 0C C3 00 02 DD 0C DC 0C 83 01 :B3
3E30 02 10 26 00 03 16 02 83 9E 18 EC 02 DD 2E 9E 16 :39
3E40 4F E6 84 83 00 3A 10 26 00 91 DC 16 C3 00 01 DD :D0
3E50 16 DC 08 83 00 02 DD 08 DC 2E 9E 08 ED 84 17 F6 :92
3E60 D8 9E 08 EC 84 DD 2E DC 08 C3 00 02 DD 08 9E 16 :3B
3E70 4F E6 84 83 00 29 10 27 00 03 16 01 B1 DC 16 C3 :1C
3E80 00 01 DD 16 DC 1A 83 00 01 10 26 00 26 CC 00 03 :99
3E90 DD 1A DC 04 83 00 03 DD 04 CC 00 9E 9E 04 E7 84 :B5
3EA0 DC 2E 9E 04 E7 01 CC 4F E6 9E 04 ED 02 16 00 A0 :DC
3EB0 16 00 28 CC 00 9E 9E 04 E7 84 DC 2E 9E 04 E7 01 :49
3EC0 CC E6 8B 9E 04 ED 02 CC 00 4F 9E 04 E7 04 DC 04 :56
3ED0 C3 00 05 DD 04 CC 00 03 DD 1A 39 9E 16 4F E6 84 :15
3EE0 83 00 28 10 26 01 29 DC 16 C3 00 01 DD 16 DC 08 :98
3EF0 83 00 02 DD 08 DC 2E 9E 08 ED 84 17 F6 3B 9E 08 :79
-----------------------------------------------------------
Sum: 2E 72 8F CE F2 D5 99 F8 60 83 BA 46 20 37 B4 3B :7E

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3F00 EC 84 DD 2E DC 08 C3 00 02 DD 08 9E 16 4F E6 84 :76
3F10 83 00 29 10 27 00 03 16 01 14 DC 16 C3 00 01 DD :A4
3F20 16 DC 1A 83 00 01 10 26 00 BE CC 00 03 DD 1A DC :26
3F30 04 83 00 04 DD 04 CC 00 9E 9E 04 E7 01 DC 2E 9E :08
3F40 04 E7 02 CC 00 EC 9E 04 E7 03 DC 24 58 49 DD 24 :D3
3F50 DC 24 83 00 00 10 26 00 0F CC 00 84 9E 04 E7 04 :A5
3F60 DC 04 C3 00 05 DD 04 39 DC 24 36 06 CC 00 10 BD :97
3F70 4D A1 36 06 DC 24 36 06 CC FF EF BD 4D A9 A4 C0 :37
3F80 E4 C0 10 27 00 17 DC 24 36 06 CC 00 1F A4 C0 E4 :61
3F90 C0 9E 04 E7 04 DC 04 C3 00 05 DD 04 39 DC 24 36 :45
3FA0 06 CC 00 80 BD 4D A1 36 06 DC 24 36 06 CC FF 7F :BF
3FB0 BD 4D A9 A4 C0 E4 C0 10 27 00 15 CC 00 88 9E 04 :FD
3FC0 E7 04 DC 24 9E 04 E7 05 DC 04 C3 00 06 DD 04 39 :3C
3FD0 CC 00 89 9E 04 E7 04 DC 04 C3 00 07 DD 04 DC 24 :6D
3FE0 9E 04 ED 1E 39 16 00 28 CC 00 9E 9E 04 E7 84 DC :77
3FF0 2E 9E 04 E7 01 CC 58 49 9E 04 ED 02 CC EC 8B 9E :97
-----------------------------------------------------------
Sum: 78 B0 B1 90 1E FB 24 FE EC F1 E5 B3 FD 86 17 F4 :A7

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4000 04 ED 04 DC 04 C3 00 06 DD 04 CC 00 03 DD 1A 39 :7E
4010 CC 00 DC 9E 04 E7 84 DC 2E 9E 04 E7 01 DC 04 C3 :EC
4020 00 02 DD 04 CC 00 02 DD 1A DC 2E DD 24 39 BD 4D :F6
4030 17 BD 4D 0B 49 4C 4C 45 47 41 4C 20 45 58 50 52 :85
4040 45 53 53 49 4F 4E 2E 00 BD 4D 0B 20 41 54 20 4C :35
4050 49 4E 45 20 00 DC 12 BD 4E B0 BD 4D 17 BD 4D 17 :E7
4060 DC 00 DD 2E 9E 2E 4F E6 84 BD 4E DE DC 2E C3 00 :22
4070 01 DD 2E 9E 2E 4F E6 1F 83 00 0D 26 E7 CC 00 0A :9F
4080 BD 4E DE DC 16 93 00 10 27 00 07 DC 16 93 00 BD :EE
4090 4E B6 BD 4D 0B 5E 00 BD 4D 17 BD 4D 17 BD 20 09 :9F
40A0 BD 4D 17 BD 4D 0B 49 4C 4C 45 47 41 4C 20 41 52 :E3
40B0 47 55 4D 45 4E 54 2E 00 16 FF BD 4D 17 BD 4D :CB
40C0 0B 54 4F 4F 20 4D 41 4E 59 20 56 41 52 49 41 42 :27
40D0 4C 45 53 2E 00 16 FF 70 DC 1A 36 06 CC 00 04 BD :56
40E0 4D B9 36 06 DC 42 36 06 CC 00 03 BD 4D A1 A4 C0 :7A
40F0 E4 C0 10 27 00 8A DC 04 83 00 05 DD 04 DC 42 83 :4F
-----------------------------------------------------------
Sum: E9 E2 94 93 F0 1C 10 A7 D8 0E 99 5D BD A2 A4 AF :43

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4100 00 01 10 26 00 10 CC 00 83 9E 04 E7 1D DC 24 9E :DA
4110 04 ED 1E 16 00 0D CC 00 93 9E 04 E7 1E DC 24 9E :D6
4120 04 E7 1F DC 2C 83 4D 99 10 26 00 08 CC 10 2D DD :9F
4130 2C 16 00 49 DC 2C 83 4D A9 10 26 00 08 CC 10 2F :55
4140 DD 2C 16 00 38 DC 2C 83 4D A1 10 26 00 08 CC 10 :EA
4150 2C DD 2C 16 00 27 DC 2C 83 4D B1 10 26 00 08 CC :05
4160 10 2E DD 2C 16 00 16 DC 2C 83 4D B9 10 26 00 08 :42
4170 CC 10 26 DD 2C 16 00 05 CC 10 27 DD 2C 16 00 05 :4D
4180 CC 10 27 DD 2C 39 DC 00 12 DD 16 9E 46 EC 84 DD :57
4190 44 9E 46 EC 02 DD 12 DC 44 83 00 00 10 26 00 08 :E6
41A0 CC 7F 0D 9E 16 ED 84 39 DC 46 C3 00 06 DD 46 DC :A0
41B0 46 93 44 10 2C 00 1A 9E 46 4F E6 84 9E 16 E7 84 :2F
```

拡張Kコンパイラ　ダンプ・リスト

```
41C0 DC 16 C3 00 01 DD 16 DC 46 C3 00 01 DD 46 16 FF :C7
41D0 DE CC 00 0D 9E 16 E7 1F DC 44 DD 46 DC 00 12 DD :7F
41E0 16 39 CC 00 33 DD 46 9E 46 EC 84 DD 46 39 39 39 :93
41F0 39 35 29 3D 24 44 44 20 59 3A 36 29 3D 53 28 00 :4A
---------------------------------------------------------
Sum: 44 42 08 41 E8 FC 99 E2 D0 15 B9 11 A7 AF 93 8B :51

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4D00 7E 4F 9B 7E 4F 80 BD 4F B5 39 0C 35 10 20 02 8D :AF
4D10 EF A6 80 26 FA 6E 84 86 0D 8D E5 86 0A 20 E1 36 :F3
4D20 06 A6 43 3D 34 06 EC 41 3D EB E4 E7 E4 A6 C4 E6 :BA
4D30 43 3D EB E0 A6 E0 1E 89 33 44 39 8E 00 00 34 10 :FA
4D40 34 10 34 10 4D 2A 04 8D 40 63 61 6C 60 2B 40 58 :23
4D50 49 2A F8 44 56 ED 64 EC C1 2A 04 8D 2C 63 61 6A :18
4D60 60 2B 1E 58 49 2A F8 44 56 A3 64 20 06 58 49 24 :F8
4D70 F8 E3 64 1C FE 2B 02 1A 01 69 63 69 62 6A 60 2A :2C
4D80 EC EC 62 6D 61 32 66 2A 05 43 53 C3 00 01 39 32 :94
4D90 66 CC 7F FF 39 4D 2B F1 39 A3 C1 2F 2C CC 00 00 :16
4DA0 39 A3 C1 2E 24 CC 00 00 39 A3 C1 2D 1C CC 00 00 :6D
4DB0 39 A3 C1 2C 14 CC 00 00 39 A3 C1 27 0C CC 00 00 :45
4DC0 39 A3 C1 26 04 CC 00 00 39 CC 00 01 39 83 00 00 :55
4DD0 27 F7 CC 00 00 39 17 FF 2A 1F 89 4F 39 8D 15 16 :4B
4DE0 FF A7 8E 04 3D 17 FF 1E A6 80 81 24 27 2B 81 2D :74
4DF0 27 EB 30 1F CC 00 00 34 06 A6 84 82 30 2B 18 81 :07
---------------------------------------------------------
Sum: D5 4A A5 98 EC 73 54 E2 49 CB 5E EE 0F 01 0C BF :2C

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4E00 09 22 14 A7 84 EC 60 58 49 58 49 E3 60 58 49 EB :C7
4E10 80 89 00 ED 60 20 E2 35 86 CC 00 00 34 06 A6 84 :43
4E20 82 30 2B F3 81 09 23 06 82 07 81 0F 22 E9 A7 84 :D2
4E30 EC 60 58 49 58 49 58 49 58 49 EB 80 ED 60 20 DE :86
4E40 8E 00 00 36 10 36 10 36 10 36 10 4D 2A 05 63 40 :C5
4E50 17 00 D7 58 49 ED 46 8E 00 0F C6 05 1C FE A6 C5 :AF
```

```
4E60 49 81 09 22 04 1C FE 20 04 82 0A 1A 01 A7 C5 5A :A4
4E70 26 EC 68 47 69 46 24 02 6C 45 30 1F 26 DC C6 01 :5F
4E80 A6 C5 26 05 5C C1 05 26 F7 39 8D B4 34 04 EB 49 :BB
4E90 C2 06 23 02 8D 20 35 04 6D 40 2A 05 86 2D 17 FE :77
4EA0 5F A6 C5 8B 30 17 FE 58 5C C1 06 26 F4 33 4A 39 :E5
4EB0 33 5E 8D 8C 20 E2 86 20 17 FE 45 5A 26 FA 39 34 :93
4EC0 04 8D 04 35 04 1F 98 1F 89 44 44 44 44 8D 04 1F :ED
4ED0 98 84 0F 8B 30 81 39 23 02 8B 07 16 FE 22 1F 98 :44
4EE0 20 F9 33 5C A6 45 E6 47 3D ED 42 EC 45 3D EB 42 :C7
4EF0 89 00 ED 41 A6 44 E6 47 3D E3 41 ED 41 A6 44 E6 :2D
---------------------------------------------------------
Sum: 4A 81 AD 42 3C E6 90 34 05 57 95 69 AC 1D 21 C4 :A8

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4F00 46 3D EB 41 89 00 ED C4 39 34 06 EC 8C 1A 8E 3D :B9
4F10 09 36 16 8D CD EC 42 33 48 C3 00 03 ED 8C 09 35 :D5
4F20 10 36 16 8D BD 33 48 39 1C 2C 6A 49 16 FE 5A 20 :E3
4F30 7C 4F 34 3B 00 00 00 00 00 00 00 00 00 00 00 00 :3A
4F40 00 00 FF FF 00 00 FF FF 00 00 FF FF 00 00 FF FF :F8
4F50 00 00 FF FF 00 00 FF FF 00 00 FF FF 00 00 FF FF :F8
4F60 00 00 FF FF 00 00 FF FF 00 00 FF FF 00 00 FF FF :F8
4F70 00 00 FF FF 00 00 FF FF 00 00 FF FF 00 00 FF FF :F8
4F80 34 7D 86 00 1F 8B 86 01 97 BF BD D3 36 35 FD FB :B1
4F90 FA 25 06 6D 07 27 F6 A6 06 35 FD 34 7F C6 00 1F :2C
4FA0 9B C6 02 D7 BF BD D3 52 35 FF CC 00 01 ED 04 AD :7A
4FB0 9F FB FA 35 FF 34 6F 4F 1F 8B C6 01 D7 BF BD DA :58
4FC0 50 25 13 0D C0 26 0F 8E 04 3D A6 80 81 00 26 FA :20
4FD0 86 0D A7 82 20 08 CC 7F 0D 8E 04 3D ED 84 8E 04 :0E
4FE0 3D 35 EF 00 00 00 00 00 00 00 00 00 00 00 00 00 :61
4FF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
---------------------------------------------------------
Sum: 56 C2 78 9A D7 F0 0C 81 9F 6C 62 F9 8A CF 5F 2D :C9
```

RANDOM BOX　FM-8用チェック・サム プログラム

I/Oエンジンルーム

プログラムを“RUN”させると，スタート・アドレス，エンド・アドレスを聞いてくるので，それぞれ16進数で入力してください．

チェック・サムの見方ですが，横の“：”の次の“Sum”はその一行の合計の下2桁で，ブロックの一番下の線で区切られた“Sum”は，縦1列の合計の下2桁です．

途中で止めたいときは“ESC”キーを，続きは他のキーをそれぞれ押してください．

```
0 CLEAR 200,&H1FFF:DIM SUM(15)
10 DEFFNCH$(E,P)=RIGHT$(STRING$(E,"0")+HEX$(P),E)
20 INPUT "Start add";S$:S=VAL("&H"+LEFT$(S$,LEN(S$)-1)+"0")
30 INPUT"End  add";E$:E=VAL("&H"+E$)
40 FOR A=S TO E STEP 16:SUM=0:IF RIGHT$(HEX$(A),2)="00" OR A=S THEN GOSUB 90
50 P$=FNCH$(4,A)+" "
60 FOR I=0 TO 15:D=PEEK(I+A):P$=P$+FNCH$(2,D)+" ":SUM=SUM+D:SUM(I)=SUM(I)+D:NEXT
70 P$=P$+":"+FNCH$(2,SUM):GOSUB 200:NEXT:GOSUB 110:CLOSE:END
90 IF A=S THEN 100 ELSE GOSUB110
100 P$=CHR$(13)+"Add  ":FOR I=0 TO 15:P$=P$+"+"+HEX$(I)+" ":NEXT:P$=P$+"Sum":GOT
O 200
110 P$=STRING$(57,"-"):GOSUB 200
120 P$="Sum:":SUM=0:FOR I=0 TO 15:P$=P$+" "+FNCH$(2,SUM(I)):SUM=SUM(I)+SUM:SUM(I
)=0:NEXT:P$=P$+" :"+FNCH$(2,SUM)+CHR$(10)+CHR$(10):SUM=0:GOTO 200
200 PRINT P$:RETURN
```

FM-8

スクリーン・アセンブラ

■アニーバンネス

便利なスクリーン・エディタによるカセットテープ・ベースの6809セルフアセンブラです．2パス型式でラベル・シンボルが使えるほか，2×2パスにより複数のファイルのリンクが可能です．

★セレクト・スイッチ

アセンブラ（エディタ）の他に逆アセンブラ，モニタなどのユーティリティが同時に常駐しており，表示される8種の機能のうちいずれかを選択します．セレクト・スイッチに戻るときは PF・1 キーを使います．PF・2 ～ PF・10 はユーザーに開放されており，あらかじめ〝LDA〞などを入れておくとエディタで利用できます．

★モニタ

〝Monitor？〞に対しアドレスを入力します．オプションが2通りあり〝P〞はプリンタ出力．〝X〞．〝Y〞．〝Z〞は出力行数がそれぞれ8，4，2となります．

カーソル移動でメモリ内容の変更を行なうのですが，ESC によりモードのASCII/16進切り換えができます．

★逆アセンブラ

モニタ同様，アドレスとオプション（〝P〞のみ）を入力します．ブランチに対しては行き先番地も表示します．問題はアドレシングの記述法で，**表1**のように多少風変わりな面もありますが，次の原則に従っています．

❶数値は以下の例外を除いて16進数で記述し，前後に〝$〞，〝H〞などのキャラクタは付けない．

例外1）8，16ビット・オフセットに限り〝$〞をプレフィクスとする．

例外2）5ビット・オフセットに限り符号つき10進数を用い〝%〞をプレフィクスとする．

❷数値またはシンボルが〝#〞に続いたときには即値演算数を，裸で表れたときはアドレスを，また，〝<>〞で囲まれたときはその内容がアドレスを表す．

表1　アドレッシング・モードの記述法

		（記述例）
アドレス・バスに関与しないもの	固有モード (inherent)	RORA：ABX：SWI2：INCB etc.
	レジスタ・モード	PSHS X,Y：TFRD-PC など計6種
	イミディエイト・モード	ADDA #OA：CMPD#symb etc.
アドレス・バスに関与するもの	ダイレクト・ページ・モード	LDA <P/OE：TST<P/sym etc.
	エクステンド・ダイレクト・モード	SUBD 5CIE：INC symb etc.
	インデクス・モード　エクステンド・インダイレクト	STD<5CIE>：LSL <symb> etc.
	インデクス・モード　インデクス	LDD<X>：STD<S>$02 etc.
	インデクス・モード　インデクス・インダイレクト	ORA<<X>>：STA<<Y>$2A>etc.

★エディタおよびアセンブラ

ファイルはV-RAM風に絶対番地にシーケンシャルに割り付けられており，したがってステップ数一定（100）になっています．

アドレッシングの記述は逆アセンブラと基本的には同じで，ラベルは〝@〞に，コメントは〝*〞に続けます．また左側7文字分はラベルに使います．複数のファイルをリンクするには次の手順を2回行ないます．

```
        〔LOAD〕
        ┌ ORGN_hhhh
ファイル1．│   ⋮
        └ CONT
〔アセンブル〕 〔転送〕
        〔LOAD〕
        ┌ ORGN_CONT
ファイル2．│   ⋮
        └ CONT
〔アセンブル〕 〔転送〕
           ⋮
```

2回目において〝Undefined Label〞がすべて0になればOKです．転送は〝Transferor〞で**Y**を2度押します．

☆おわりに

書き尽せなかったところもありますが，エラーに対してはタフに作ったつもりですので試行錯誤でやってみてください．自分で打ち込む場合，BASIC の100―2000および9000―ができると Monitor が走ります．すると添付リストと同じ型式でリストができますから多少手間がはぶけると思います．

プリンタがない場合，第1行目を〝100 GOTO 500〞に変更してください．

◘参考文献

1）F-BASIC マニュアル

2）ASCII 出版：6809ハンドブック

3）I/O 誌 関連記事

（表2）インデックス・モードのオプション

オプション	インデックス	インデックス・インダイレクト
0オフセット	〈R〉	〈〈R〉〉
ポストインクリメント 1	〈R〉+	
〃 2	〈R〉++	〈〈R〉〉++
プレデクリメント 1	−〈R〉	
〃 2	−−〈R〉	−−〈〈R〉〉
5ビット・オフセット	〈R〉%d	
8ビット・オフセット	〈R〉$hh	〈〈R〉$hh〉
16ビット・オフセット	〈R〉$hhhh	〈〈R〉$hhhh〉
A オフセット	〈R〉'A	〈〈R〉'A〉
B オフセット	〈R〉'B	〈〈R〉'B〉
D オフセット	〈R〉'D	〈〈R〉'D〉
PC相対8ビット・オフセット	〈PC〉$hh	〈〈PC〉$hh〉
PC相対16ビット・オフセット	〈PC〉$hhhh	〈〈PC〉$hhhh〉

R……ポインタ・レジスタ X，Y，U，S

h……16進数 0，1，……，F

d……符号付き10進数 −16〜+0〜+15

アセンブラ擬似命令

SET……シンボル定義 例：SET /カウンタ=2F

SET Wdata=5C6E

ORGN……オブジェクトのベース設定

例：ORGN 68A0

ORGN CONT

CONT……現在のオブジェクトのPCを&H3C88にストア

END……以下のステップを無視

LCLR……シンボル/ラベル・テーブルをクリア

(inherent mode) tn……n回連続命令（1〜6回）

例 LSLA t4

マシン語エリア・マップ

アドレス	エリア
3A40	
↓	メイン・プログラム
3C00	
↕	ダイレクト・ページ
↑	スタック
3D00	
↓	サブルーチン群
4000	
↓	プログラム群
4970	
↓	テーブル
5000	
↓	ソース・ファイル
5C1C	
5C20	
↓	オブジェクト
5D80	
↓	シンボル・エリア
5E00	
↓	ラベル・エリア
6000	
↕	フリーエリア

主なレジスタ

（DP：&H3C内）

10……転送開始番地

12……〃 終了番地

14……転送先番地

96……クリア開始番地

98……〃 終了番地

80……Orign レジスタ

88……CONT レジスタ

8F……エラーフラグ

90……Undofined Label

91……CRT出力ライン

93……プリンタ出力ライン

（DP外）

5C1E……オブジェクト・ポインタ

ASCIIコード表（エディタ）

特殊キー	CTRL	キャラクタ	Hexコード	機　能
	A	S_H	0 1	アセンブル（リストなし）
	B	S_X	0 2	アセンブル（リスト→CRT）
	C	E_X	0 3	ソース・ファイル・クリア
	D	E_T	0 4	ライン・デリート
〔EL〕	E	E_Q	0 5	(dead)
〔BS〕	H	B_S	0 8	$H_X \to 9$
〔TAB〕	I	H_T	0 9	ライン・インサート
〔HOME〕	K	H_M	0 B	$H_X \to 16$
〔CLEAR〕	L	C_L	0 C	CRT クリア
〔RETURN〕	M	C_R	0 D	$H_X \to 16$: $H_Y \to H_Y + 1$
	P	D_E	1 0	アセンブル（リスト→プリンタ）
〔DUP〕	Q	D_1	1 1	(dead)
〔INS〕	R	D_2	1 2	(dead)
〔ESC〕		E_C	1 B	(dead)
〔DEL〕		D_L	7 F	(dead)
		→	1 C	カーソル移動
		←	1 D	〃
		↑	1 E	〃
		↓	1 F	〃
		その他１F以下のコード		(dead)
		¥	5 C	"$"
		∧	5 E	"#"
		〔	3 C	"<"
		〕	3 E	">"

H_X, H_Y：カーソル座標

ラベルおよびシンボル

ラベル……"@"に続く英数６文字以内（ただし先頭は英字）.

２バイト・シンボル……英数６文字以内（ただし先頭は英小文字）.

１バイト・シンボル……"/"に続く英数６文字以内（ただし先頭にA～Fは除く）.

テーブル

Index Post Byee	4970～4A6F
Code Format	4A70～4B0F
Page 3	4B10～4B3D
Page 2	4B42～4BFF
Page 1	4C00～4FFF

スクリーン・アセンブラ　ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3A40 10 FF 3C BE 10 CE 3C BE 86 3C 1E 8B 97 C0 0F 8E :40
3A50 0F 8F CC 5C 20 FD 5C 1E 0F 40 0C 40 BD 42 38 0D :3C
3A60 6B 27 05 86 02 16 00 AE 0D 6C 27 05 86 03 16 00 :27
3A70 A5 0D 6D 27 05 86 04 16 00 9C 0D 6E 27 05 86 05 :B9
3A80 16 00 93 0D 6F 27 05 86 06 16 00 8A 0D 4A 27 13 :0E
3A90 BD 47 F0 0D 9E 27 04 86 0B 20 7B 0D 9F 27 04 86 :53
3AA0 0C 20 73 0D 52 27 14 BD 43 80 0D 72 10 27 00 A1 :10
3AB0 BD 44 98 0D 6A 27 04 86 01 20 5B 0D 8E 27 39 0D :45
3AC0 64 27 13 BD 48 43 0D 9D 27 04 86 0A 20 48 0D 9F :5F
3AD0 27 04 86 0C 20 40 96 40 91 91 2B 04 91 92 2B 68 :FA
3AE0 91 93 2B 14 91 94 2B 60 20 0E 10 FF 3C BE 10 CE :28
3AF0 3C BE 86 3C 1E 8B 97 C0 FC 5C 1E 10 83 5E 00 2B :4E
-----------------------------------------------------------
Sum: 23 E9 52 14 17 A5 22 EC CB 59 20 71 BB BF 8F E7 :E1


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3B00 04 86 09 20 41 86 63 91 40 10 2A FF 4D 0D 8E 26 :F5
3B10 37 0C 8E 16 FF 3C 81 01 26 2C DC 50 10 83 4A 4D :4C
3B20 27 0E 10 83 4A 53 27 04 86 01 20 1A C6 BD 20 02 :F6
3B30 C6 7E 34 10 BE 5C 1E E7 80 CC FF FF ED 81 BF 5C :7A
3B40 1E 35 10 16 FF 75 97 8F 96 C0 1F 8B 10 FE 3C BE :1B
3B50 39 DC 50 10 83 4F 52 26 38 DC 52 10 83 47 4E 26 :73
3B60 54 0D 8E 26 0C DC 56 10 83 4E 54 26 06 DC 88 DD :F5
3B70 80 20 40 DC 54 BD 3E E3 5D 27 04 86 01 20 C7 97 :7B
3B80 80 DC 56 BD 3E E3 5D 27 04 86 01 20 F0 97 81 20 :E7
3B90 E0 10 83 43 4F 26 14 DC 52 10 83 4E 54 26 16 FC :DA
3BA0 5C 1E 83 5C 20 D3 80 DD 88 20 C6 10 83 45 4E 26 :63
3BB0 08 86 64 20 3E 86 0D 20 C4 10 83 53 45 26 15 BD :EA
3BC0 41 90 0D 1E 27 05 86 07 16 FF B2 0D 1F 27 E4 86 :39
3BD0 08 16 FF A9 10 83 4C 43 26 DB DC 52 10 83 4C 52 :48
3BE0 26 D3 0D 8E 26 CD CC 5D 80 DD 96 CC 5F FF DD 98 :42
3BF0 BD 48 D4 0D 8E 10 27 FE FF 16 FE DA 00 00 00 00 :96
-----------------------------------------------------------
Sum: 43 AD B6 CF 00 95 69 CA 77 AD DD 85 44 E0 97 98 :16


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3C00 00 01 01 01 01 00 00 01 7F 43 3C 71 00 00 00 00 :74
3C10 3C 5B 3C 63 3C 5A 00 08 00 09 5C 29 48 50 00 00 :FA
3C20 00 05 00 39 4A 70 05 00 4A 93 00 00 00 03 00 17 :F4
3C30 00 00 00 00 4B F2 9E 00 4B A4 00 0C 00 00 00 00 :D6
3C40 64 5A 14 00 00 4A 50 54 64 00 00 00 00 00 00 00 :24
3C50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
3C60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
3C70 00 3F 00 00 00 00 00 00 00 00 00 00 04 00 00 00 :43
3C80 7F 00 00 45 37 00 00 00 7F 78 5C 90 00 00 01 00 :DF
3C90 00 00 15 00 5B 00 5F FF 5F FF 00 00 00 00 00 00 :2C
3CA0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 A0 43 :E3
3CB0 F1 3C 4C B1 00 C5 3C F2 3D 36 43 7D 3A 5F 38 AC :CD
3CC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
3CD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
3CE0 00 00 00 00 00 00 00 00 00 00 00 00 00 4B AE 4C :45
3CF0 E8 45 4E 44 20 63 98 3C 3A EA 00 C5 3C BA EB A8 :88
-----------------------------------------------------------
Sum: F8 7B 00 D7 84 2E 26 8A CD 1A 37 78 C2 B7 72 FA :27


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3D00 FF 3C FE CE 3C FE 36 7E 86 3C 1F 8B 39 CE 3C F5 :99
3D10 37 7E FE 3C FE 39 12 12 0F 18 0F 19 DC 12 93 10 :2A
3D20 2A 01 39 DD 16 DC 10 93 14 2B 14 9E 10 10 9E 14 :99
3D30 A6 80 A7 A0 8D 24 DC 16 93 18 2B 02 20 F2 39 9E :D1
3D40 12 DC 14 D3 16 1F 02 6D 80 6D A0 A6 82 A7 A2 8D :04
3D50 09 DC 16 93 18 2B 02 20 F2 39 0C 19 26 02 0C 18 :8F
3D60 39 12 12 12 96 26 D6 23 3D DD 28 96 26 D6 22 3D :57
3D70 D3 27 DD 27 86 03 97 2F 04 27 06 28 06 29 06 2A :05
3D80 0A 2F 26 F4 86 05 97 2F 04 2A 0A 2F 26 FA DC 28 :2F
3D90 D3 24 DD 28 12 10 9E 28 86 08 97 2E 90 2A 97 2D :B5
3DA0 86 08 90 26 97 2C 96 26 97 2F 96 20 26 27 A6 A4 :D6
3DB0 0D 2A 27 07 44 0A 2E 0A 2A 26 F9 44 56 0A 2F 27 :2E
3DC0 08 0A 2E 26 F6 A6 21 20 F2 0D 2C 27 05 54 0A 2C :24
3DD0 26 FB D7 21 39 12 96 2A 97 2E EC A4 1E 89 0D 2A :57
3DE0 27 08 44 56 06 2B 0A 2A 20 F4 04 21 06 2D 44 56 :34
3DF0 0A 2F 26 F6 08 2D 59 49 0A 26 26 F8 0D 2E 27 08 :E4
-----------------------------------------------------------
Sum: FC ED 1E 02 D7 05 B8 5C ED 1D B9 66 81 17 46 97 :97


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3E00 08 2B 59 49 0A 2E 20 F4 1E 89 ED A4 39 12 12 12 :C8
3E10 CC 3C 08 DD 18 A6 9F 3C 08 0A 19 6F 9F 3C 18 48 :5B
3E20 24 04 6C 9F 3C 18 0D 19 26 EF 39 44 56 44 56 44 :73
3E30 56 44 56 54 54 54 54 81 0A 2A 04 8B 30 20 02 8B :61
3E40 37 C1 0A 2A 03 CB 30 39 CB 37 39 86 1F 97 2F D6 :DF
3E50 40 5A 3D C3 50 00 1F 01 C3 00 07 DD 41 86 20 A7 :3F
3E60 80 0A 2F 26 FA 86 2A A7 9F 3C 41 39 86 04 97 2F :D5
3E70 0F 7C 37 02 A1 80 27 02 0C 7C 0A 2F 26 F4 1F 30 :38
3E80 83 00 04 1F 03 39 81 30 2B 17 81 3A 2A 01 39 81 :75
3E90 41 2B 13 81 5B 2A 01 39 81 61 2B 0A 81 7B 2A 01 :FD
3EA0 39 81 00 2A 01 39 4F 39 81 2F 27 2E 81 30 2B 30 :B7
3EB0 81 3A 2B 29 81 41 2B 28 81 47 2B 21 81 53 27 17 :4A
3EC0 81 55 27 13 81 58 27 0F 81 59 27 0B 81 61 2B 10 :48
3ED0 81 7B 2A 0C C6 03 39 C6 04 39 C6 02 39 C6 01 39 :38
3EE0 C6 00 39 81 41 2B 04 80 37 20 02 80 30 C1 47 2A :AB
3EF0 1F C1 41 2B 04 C0 37 20 0A C1 3A 2A 13 C1 30 2B :C5
-----------------------------------------------------------
Sum: B9 C7 DD EC 0C 34 57 EC 03 FC F5 F7 14 6F DF 6C :85


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3F00 0F C0 30 58 58 58 58 58 49 58 49 58 49 58 49 39 :1C
3F10 C6 01 39 4F E6 A0 C1 41 26 03 C6 08 39 C1 42 26 :30
3F20 03 C6 09 39 C1 58 26 03 C6 01 39 C1 59 26 03 C6 :56
3F30 02 39 C1 55 26 03 C6 03 39 C1 53 26 03 C6 04 39 :BC
```

```
3F40 C1 43 26 09 E6 A0 C1 43 26 27 C6 0A 39 C1 50 26 :4A
3F50 09 E6 A0 C1 43 26 1A C6 05 39 C1 44 26 13 E6 A0 :9B
3F60 27 04 C1 2D 26 04 6D A2 5F 39 C1 50 26 03 C6 0B :F5
3F70 39 4C 39 4F E6 A0 C1 41 26 03 C6 02 39 C1 42 26 :E8
3F80 03 C6 04 39 C1 58 26 03 C6 10 39 C1 59 26 03 C6 :60
3F90 20 39 C1 43 26 09 E6 A0 C1 43 26 2A C6 01 39 C1 :27
3FA0 44 26 09 E6 A0 C1 50 26 1D C6 08 39 C1 55 26 09 :99
3FB0 E6 A0 C1 53 26 10 C6 40 39 C1 50 26 09 E6 A0 C1 :96
3FC0 43 26 03 C6 80 39 4C 39 0F 0E 1F 30 CB 06 1F 03 :CF
3FD0 86 06 97 2F A6 C2 A1 80 27 02 0C 0E 0A 2F 26 F4 :71
3FE0 39 91 72 26 03 DC 70 39 91 75 26 03 DC 73 39 91 :32
3FF0 78 26 03 DC 76 39 91 7B 26 03 DC 79 39 86 03 39 :B1
-----------------------------------------------------------
Sum: CB E1 91 27 A6 FF 1E 01 E8 1B 8D EB 6F 2D 53 67 :F9


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4000 BD 3D 00 BD 3D 18 BD 3D 0D 39 BD 3D 00 86 05 97 :68
4010 26 CC 4A 70 DD 24 EC 02 DD 22 0F 20 BD 3D 64 6F :96
4020 02 96 21 A7 03 BD 3D 0D 39 BD 3D 00 96 36 C6 04 :33
4030 3D C3 4C 00 DD 38 10 AE 01 EC 9F 3C 38 ED 20 DC :08
4040 38 C3 00 02 DD 38 EC 9F 3C 38 ED 22 BD 3D 0D 39 :60
4050 BD 3D 00 10 AE 01 CC 4B DA DD 34 CC 4B 42 DD 38 :29
4060 86 26 97 2F A6 9F 3C 34 90 36 27 1D DC 34 C3 00 :04
4070 01 DD 34 DC 38 C3 00 04 DD 38 0A 2F 26 E6 CC 20 :33
4080 20 ED 20 ED 22 BD 3D 0D 39 EC 9F 3C 38 ED 20 DC :64
4090 38 C3 00 02 DD 38 EC 9F 3C 38 20 E7 BD 3D 00 10 :22
40A0 AE 01 CC 4B 34 DD 34 CC 4B 10 DD 38 86 09 97 2F :9C
40B0 A6 9F 3C 34 90 36 27 1D DC 34 C3 00 01 DD 34 DC :80
40C0 38 C3 00 04 DD 38 0A 2F 26 E6 CC 20 20 ED 20 ED :5F
40D0 22 BD 3D 0D 39 EC 9F 3C 38 ED 20 DC 38 C3 00 02 :47
40E0 DD 38 EC 9F 3C 38 20 E7 BD 3D 00 BD 3E 10 BD 3D :1A
40F0 0D 39 BD 3D 00 0F 54 10 AE 01 EC A4 36 06 96 3B :FF
-----------------------------------------------------------
Sum: 8E A6 90 4C 78 3F 8B 13 0C 00 31 8B DD 55 26 D5 :5A


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4100 C6 08 3D C3 49 70 1F 01 86 08 97 2E 86 0B 97 2F :51
4110 A6 80 81 53 27 1A 81 57 27 2B A7 A0 0A 2F 27 0C :18
4120 0A 2E 26 EC 86 20 A7 A0 0A 2F 26 F8 BD 3D 0D 39 :CE
4130 37 02 BD 3E 2B A7 A0 0A 2F E7 A0 0C 54 0A 2E 0A :08
4140 2F 27 E9 20 CB 37 02 BD 3E 2B A7 A0 0A 2F E7 A0 :90
4150 0C 54 0A 2F 20 DA BD 3D 00 86 64 97 40 BD 3E 4B :94
4160 0A 40 26 F9 86 20 B7 5C 1C BD 3D 0D 39 BD 3D 00 :78
4170 86 10 97 2F 86 1F D6 40 5A 3D C3 50 00 1F 02 EC :CE
4180 01 1F 01 EC A1 ED 81 0A 2F 26 F8 BD 3D 0D 39 12 :C5
4190 BD 3D 00 0F 1E 0F 1F 8E 5D 80 10 8E 3C 54 A6 A4 :38
41A0 81 2F 26 6C 6D A0 A6 A4 BD 3E 86 27 2E BD 3E A8 :12
41B0 C1 01 27 27 1F 50 DD 1C 7E 49 38 0D 0D 26 1C A6 :79
41C0 80 27 1C 81 2F 26 07 BD 3F C8 0D 0E 27 28 6D 80 :BB
41D0 8C 5E 00 2B EA 0C 1E BD 3D 0D 39 0C 1F 20 F8 86 :32
41E0 2F A7 1F 1F 30 C3 00 06 1F 03 86 06 97 2F A6 C2 :E9
41F0 A7 80 0A 2F 26 F8 A6 A0 BD 3E A8 C1 01 26 DC E6 :11
-----------------------------------------------------------
Sum: 5A BB E4 3F D2 7A 21 10 B9 37 49 C6 B6 2A 7D 07 :18


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4200 A0 BD 3E E3 5D 26 D4 A7 80 1F 10 54 25 E8 20 C7 :73
4210 A6 A4 BD 3E A8 C1 03 26 C2 1F 50 DD 1C 7E 49 38 :00
4220 0D 0D 26 B7 A6 84 27 BB BD 3F C8 0D 0E 27 C7 6D :3D
4230 81 8C 5E 00 2B EE 20 9D BD 3D 00 8E 3C 4A 6F 80 :3E
4240 8C 3C 70 2B F9 86 07 97 2E 86 1F 97 2F D6 40 5A :89
4250 3D C3 50 00 1F 01 86 4A 97 45 86 50 97 46 86 54 :A9
4260 97 47 86 64 97 48 0D 2E 27 2E A6 80 0A 2E 0A 2F :CE
4270 81 40 26 F2 0D 2E 27 20 A6 80 0A 2E 0A 2F 81 20 :93
4280 27 F2 BD 3E 86 4D 10 27 00 DE D6 45 1E 8B 1F 02 :E1
4290 1E 8B A7 A4 0C 45 20 DC 0D 2F 10 27 00 DC A6 80 :B6
42A0 0A 2F 81 20 27 F2 81 2A 10 27 00 CE 81 41 10 2B :A0
42B0 00 BA 81 5B 10 2A 00 B4 20 20 0D 2F 10 27 00 BA :F1
42C0 A6 80 0A 2F 81 2A 10 27 00 B0 81 40 27 72 81 20 :EC
42D0 27 3A 81 41 2B 18 81 5B 2A 32 D6 46 C1 54 10 2A :09
42E0 00 96 1E 8B 1F 02 1E 8B A7 A4 0C 46 20 CC 81 32 :45
42F0 27 04 81 33 26 16 D6 50 C1 53 26 10 D6 51 C1 57 :CA
-----------------------------------------------------------
Sum: F8 3A 7B E4 4C 5E 15 92 1D 60 F9 A6 F2 02 98 23 :AD


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4300 26 0A 20 D6 0D 2F 27 72 A6 80 0A 2F 81 20 27 F4 :16
4310 81 2A 27 66 81 40 27 28 81 23 2B 0C 81 2E 2B 0E :0B
4320 81 2F 2B 44 81 3F 2B 06 BD 3E 86 4D 27 3A D6 47 :5C
4330 C1 64 2A 40 1E 8B 1F 02 1E 8B A7 A4 0C 47 20 C4 :84
4340 0D 2F 27 36 A6 80 0A 2F 81 20 27 F4 81 2A 27 2A :B0
4350 BD 3E 86 4D 27 12 D6 48 C1 6A 2A 14 1E 8B 1F 02 :58
4360 1E 8B A7 A4 0C 48 20 D8 0C 6F 20 0E 0C 6E 20 0A :8D
4370 0C 6D 20 06 0C 6C 20 02 0C 6B BD 3D 0D 39 12 12 :14
4380 BD 3D 00 96 53 27 04 36 02 20 04 86 20 36 02 96 :DE
4390 52 26 04 BD 3D 0D 39 36 02 96 51 36 02 96 50 36 :2F
43A0 02 86 10 97 2F 8E 3C 70 6F 80 0A 2F 26 FA CC 3C :E8
43B0 70 DD 0A 8E 4C 00 BD 3E 6C 0D 7C 26 32 36 10 0C :CB
43C0 0B 96 7D A7 9F 3C 0A 86 05 97 26 CC 4A 70 DD 24 :79
43D0 E6 9F 3C 0A D7 23 0F 22 0F 20 BD 3D 64 0C 0B 96 :30
43E0 21 A7 9F 3C 0A 37 10 0C 0B D6 0B C1 7C 2A A4 0C :03
43F0 7D 8C 50 00 2B C0 8E 4B 42 10 8E 4B DA 0C 0B A6 :DF
-----------------------------------------------------------
Sum: ED 5A D6 52 C8 97 A5 0C 9C B0 E7 A5 6B D9 85 D5 :F5


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4400 A0 A7 9F 3C 0A BD 3E 6C 0D 7C 26 34 0A 0B 6C 9F :96
4410 3C 0A 36 30 0C 0B 86 05 97 26 CC 4A 70 DD 24 E6 :78
4420 9F 3C 0A D7 23 0F 22 0F 20 BD 3D 64 0C 0B 96 21 :6B
4430 A7 9F 3C 0A 37 30 0C 0B 0C 0B D6 0B C1 7C 2A 54 :BD
4440 8C 4B DA 2B BA 8E 4B 10 10 8E 4B 34 A6 A0 A7 9F :28
4450 3C 0A BD 3E 6C 0D 7C 26 36 0A 0B 86 02 A7 9F 3C :B1
4460 0A 36 30 0C 0B 86 05 97 26 CC 4A 70 DD 24 E6 9F :DB
4470 3C 0A D7 23 0F 22 0F 20 BD 3D 64 0C 0B 96 21 A7 :73
4480 9F 3C 0A 37 30 0C 0B 0C 0B D6 0B C1 7C 2A 05 8C :53
4490 4B 34 2B B8 BD 3D 0D 39 BD 3D 00 BE 5C 1E 96 72 :DC
44A0 81 05 26 40 96 54 26 04 86 01 20 13 81 74 26 30 :05
44B0 D6 55 C1 31 2B 2A C1 37 2A 26 86 30 BD 3E E3 97 :E5
44C0 2F 96 70 26 0A 96 71 A7 80 0A 2F 26 FA 20 0A 8B :A1
44D0 0F D6 71 ED 81 0A 2F 26 FA BF 5C 1E BD 3D 0D 39 :96
44E0 0C 6A 20 F8 10 8E 3C 54 81 0A 26 45 96 71 81 1E :58
44F0 27 04 81 1F 26 1F BD 3F 13 4D 26 E4 86 10 3D D7 :20
-----------------------------------------------------------
Sum: E2 C5 57 6F 1F 5E 65 58 7F 65 91 52 C0 48 16 99 :25


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4500 0C A6 A0 81 2D 26 D9 BD 3F 13 4D 26 D3 DA 0C 96 :D0
4510 71 ED 81 20 C4 0F 0C BD 3F 73 4D 26 C3 DA 0C D7 :40
4520 0C A6 A0 27 06 81 2C 27 EE 20 B5 96 71 ED 81 20 :AB
4530 A8 81 02 26 2E D6 71 96 70 26 04 E7 80 20 04 86 :07
4540 10 ED 81 96 50 81 4C 27 0A 0D 54 26 1D 86 FF A7 :32
4550 80 20 09 0D 54 26 2C CC FF FF ED 81 BF 5C 1E BD :8A
4560 3D 0D 39 16 00 77 0C 6A 20 F5 10 8E 3C 54 A6 A4 :13
4570 BD 3E A8 C1 01 26 2E EC A4 BD 3E E3 5D 26 E7 A7 :38
4580 80 20 D9 10 8E 3C 54 A6 A4 BD 3E A8 C1 01 26 15 :91
4590 EC A1 BD 3E E3 5D 26 CE A7 80 A6 A4 BD 3E A8 C1 :91
45A0 01 26 C3 20 D2 9F 1A 8E 5D 80 C1 02 26 04 6D A0 :FA
45B0 6D 80 1F 50 DD 1C 7E 49 38 0D 0D 26 A9 BD 3F C8 :01
45C0 0D 0E 27 09 6D 81 8C 5E 00 2B F2 20 79 1F 12 9E :C8
45D0 1A A6 A0 A7 80 1F 20 54 25 F7 16 FF 7F 96 54 81 :35
45E0 2D 27 16 81 23 27 15 81 3C 10 26 00 78 96 55 81 :21
45F0 50 26 04 96 56 81 2F 27 42 16 00 84 86 0E BD 3F :A9
-----------------------------------------------------------
Sum: 39 7A 87 ED 50 6C 36 25 2C 9C C2 F8 5F 76 39 DF :AD


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4600 E1 81 03 27 12 4D 26 04 E7 80 20 04 8B 0F ED 81 :A8
4610 10 8E 3C 55 16 FF 57 86 13 BD 3F E1 81 03 27 12 :CE
4620 4D 26 04 E7 80 20 04 8B 0F ED 81 10 8E 3C 55 16 :4F
4630 FF 55 0C 6A BF 5C 1E BD 3D 0D 39 86 16 BD 3F E1 :BC
4640 81 03 27 EE 4D 26 04 E7 80 20 04 8B 0F ED 81 96 :39
4650 57 BD 3E A8 C1 01 26 07 10 8E 3C 57 16 FF 0F 10 :4E
4660 8E 3C 56 20 F7 86 1B BD 3F E1 81 03 27 C4 4D 26 :97
4670 04 E7 80 20 04 8B 0F ED 81 10 8E 3C 54 16 FF 07 :E1
4680 86 1D BD 3F E1 81 03 27 0D 4D 26 04 E7 80 20 0C :42
4690 8B 0F ED 81 20 06 0C 6A BD 3D 0D 39 96 55 BD 3E :CA
46A0 A8 C1 01 27 04 C1 03 26 0B 86 9F A7 80 10 8E 3C :B0
46B0 55 16 FE D3 0F 82 0F 8D 9F 8A 8E 3C 83 10 8E 3C :B9
46C0 55 A6 A0 81 25 10 27 00 BB BD 3E A8 C1 04 27 0A :CC
46D0 81 24 27 23 4D 26 EA 16 00 59 5F 81 58 27 0B 5C :81
46E0 81 59 27 06 5C 81 55 27 01 5C 58 58 58 58 58 D7 :4C
46F0 8C 86 52 A7 3F 20 CA A6 22 BD 3E A8 C1 01 27 1B :A3
-----------------------------------------------------------
Sum: 98 19 73 AE 91 A1 44 91 E8 9F FB E5 02 4A 2E 77 :31


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4700 EC A4 ED 84 0C 82 86 53 A7 A0 1F 20 DD 14 5C DD :18
4710 10 CC 3C 63 DD 12 BD 3D 18 20 18 EC A4 ED 81 EC :9E
4720 22 ED 84 0C 82 0C 82 86 57 A7 A0 1F 20 DD 14 CB :CE
4730 04 20 DC 8E 49 70 5F 10 8E 3C 54 86 08 97 2F A6 :CE
4740 A0 26 02 86 20 A1 80 27 01 5C 0A 2F 26 F1 12 12 :87
4750 12 12 5D 27 09 0C 8D 8C 4A 70 2A 24 20 D8 9E 8A :FE
4760 86 80 9A 8C 9A 8D A7 80 96 82 27 0D 10 8E 3C 83 :23
4770 81 01 10 26 FE 11 16 FD F5 BF 5C 1E BD 3D 0D 39 :48
4780 0C 6A 20 F8 A6 A0 81 2B 27 28 81 2D 26 F2 6D A8 :AA
4790 01 26 06 A6 A4 80 30 20 0A A6 A0 81 31 26 E1 A6 :F6
47A0 A4 80 26 81 11 2A D9 4D 27 D6 4A 43 84 0F 8A 10 :E3
47B0 20 1B 6D A8 01 26 06 A6 A4 80 30 20 0A A6 A0 81 :68
47C0 31 26 BD A6 A4 80 26 81 10 2A B5 84 0F 9A 8C 9E :CB
47D0 8A A7 80 BF 5C 1E BD 3D 0D 39 00 00 00 00 00 00 :2A
47E0 BD 3D 00 BD 3E 4B BD 3D 0D 39 00 00 00 00 BD 3D :7A
47F0 BD 3D 00 0F 9E 0F 9F 10 8E 3C 4A 1F 50 DD 1C 7E :5F
-----------------------------------------------------------
Sum: E1 A8 88 D8 AD C3 BD 9F 2E AC 7C E3 00 4D F6 CA :FB


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4800 49 38 0D 0D 26 39 8E 5E 00 EC 84 27 12 BD 3F C8 :53
4810 0D 0E 27 1D A4 81 8C 60 00 2B EE 0C 9E 20 1C A6 :15
4820 45 E6 44 ED 81 A6 43 E6 42 ED 81 A6 41 E6 40 ED :56
4830 81 FC 5C 1E 83 5C 20 D3 80 ED 84 BD 3D 0D 39 0C :06
4840 9F 20 F8 BD 3D 00 0F 9D 0F 9F 10 8E 3C 64 1F 50 :B8
4850 DD 1C 7E 49 38 0D 0D 27 04 0C 9F 20 59 8E 5E 00 :4D
4860 EC 84 27 0E BD 3F C8 0D 0E 27 0B A4 81 8C 60 00 :C7
4870 2B EE 0C 90 20 40 DC 50 81 4A 27 0C 81 42 27 16 :3F
4880 10 83 4C 42 27 38 20 D1 FC 5C 1E 83 00 02 1F 02 :8D
4890 EC 84 ED A4 20 20 FC 5C 1E 83 00 01 1F 02 EC 84 :CC
48A0 93 80 C3 5C 20 B3 5C 1E 10 83 FF 80 2B 06 10 83 :55
48B0 00 80 2B 06 0C 9D BD 3D 0D 39 E7 A4 20 F8 FC 5C :95
48C0 1E 83 00 02 1F 02 EC 84 93 80 C3 5C 20 B3 5C 1E :B3
48D0 ED A4 20 E2 DC 98 10 93 96 2A 01 39 4F A7 9F 3C :75
48E0 96 DC 96 10 93 98 27 F3 C3 00 01 DD 96 20 ED BD :5E
48F0 3D 00 8D E0 BD 3D 0D 39 00 00 00 00 00 00 00 00 :EA
-----------------------------------------------------------
Sum: 1C E0 E7 F5 DE 5F A2 63 87 52 21 0E 34 0C D7 49 :82


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4900 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
4910 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
4920 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
4930 00 00 00 00 00 00 00 00 0F 0D 86 06 97 2F A6 A0 :B4
4940 27 1D 81 29 27 19 81 3E 27 15 81 3D 27 11 36 02 :57
4950 BD 3E 86 26 02 0C 0D 0A 2F 26 E3 6D A0 20 08 86 :BF
4960 20 36 02 0A 2F 26 F8 DC 1C C3 00 05 1F 05 12 12 :B7
4970 3C 52 3E 2B 20 20 20 20 3C 52 3E 2B 2B 20 20 20 :F9
4980 2D 3C 52 3E 20 20 20 20 2D 2D 3C 52 3E 20 20 20 :FF
4990 3C 52 3E 20 20 20 20 20 3C 52 3E 27 42 20 20 20 :01
```

スクリーン・アセンブラ ダンプ・リスト

```
49A0 3C 52 3E 27 41 20 20 20 45 20 20 20 20 20 20 20 :B9
49B0 3C 52 3E 24 53 20 20 20 3C 52 3E 24 57 20 20 20 :4A
49C0 45 20 20 20 20 20 20 20 3C 52 3E 27 44 20 20 20 :BC
49D0 3C 50 43 3E 24 53 20 20 3C 50 43 3E 24 57 20 20 :8C
49E0 45 20 20 20 20 20 20 20 45 20 20 20 20 20 20 20 :4A
49F0 45 20 20 20 20 20 20 20 3C 3C 52 3E 3E 2B 2B 20 :E1
-------------------------------------------------------
Sum: 2C C5 F6 CB D0 9E A6 44 A0 4C F3 60 65 C7 21 5A :F0
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4A00 45 20 20 20 20 20 20 20 2D 2D 3C 3C 52 3E 3E 20 :E5
4A10 3C 3C 52 3E 3E 20 20 20 3C 3C 52 3E 27 42 3E 20 :75
4A20 3C 3C 52 3E 27 41 3E 20 45 20 20 20 20 20 20 20 :F3
4A30 3C 3C 52 3E 24 53 3E 20 3C 3C 52 3E 24 57 3E 20 :BE
4A40 45 20 20 20 20 20 20 20 3C 3C 52 3E 27 44 3E 20 :F6
4A50 3C 3C 50 43 3E 24 53 3E 3C 3C 50 43 3E 24 57 3E :00
4A60 45 20 20 20 20 20 20 20 3C 57 3E 20 20 20 20 20 :96
4A70 16 00 6B 81 B5 D6 5A 60 AD B5 00 94 02 80 10 A0 :6F
4A80 38 E0 8A 52 42 08 21 84 10 42 08 21 84 10 BD F7 :A6
4A90 AE 94 52 A0 94 E2 0A 28 05 80 52 40 29 A5 14 50 :25
4AA0 0A 28 05 80 52 40 29 A5 14 50 0A 28 1D 80 DE 41 :69
4AB0 EF BD 77 D0 7B EF 1B 80 BD C1 DE 7B 6F B0 F7 DE :C3
4AC0 CE B9 E9 9C 03 CE 39 37 C5 04 D6 5A 6B AD B5 D6 :E9
4AD0 5A 6B AD B5 BD F7 DE 7B EF BD F7 DE 7B EF 7B EF :89
4AE0 BD F7 DE 7B EF BD F7 DE CE B9 E9 9C 03 CE 39 37 :DB
4AF0 C1 04 D6 5A 6B AD B5 D6 5A 6B AD B5 BD F7 DE 7B :CC
-------------------------------------------------------
Sum: 5A C8 B3 46 99 56 DB 95 0D 01 85 9A 23 45 8C 7B :16
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4B00 EF BD F7 DE 7B EF 7B EF BD F7 DE 7B EF BD F7 DE :E3
4B10 43 4D 50 53 43 4D 50 53 43 4D 50 53 43 4D 50 53 :CC
4B20 43 4D 50 55 43 4D 50 55 43 4D 50 55 43 4D 50 55 :D4
4B30 53 57 49 33 8C 9C BC AC 83 93 B3 A3 3F 00 00 00 :61
4B40 00 00 4C 42 43 43 4C 42 43 53 4C 42 45 51 4C 42 :EA
4B50 47 45 4C 42 47 54 4C 42 48 49 4C 42 4C 45 4C 42 :81
4B60 4C 53 4C 42 4C 54 4C 42 4D 49 4C 42 4E 45 4C 42 :A0
4B70 50 4C 4C 42 52 4E 4C 42 56 43 4C 42 56 53 43 4D :B8
4B80 50 44 43 4D 50 44 43 4D 50 44 43 4D 50 44 43 4D :90
4B90 50 59 43 4D 50 59 43 4D 50 59 43 4D 50 59 4C 44 :E4
4BA0 59 20 4C 44 59 20 4C 44 59 20 4C 44 59 20 4C 44 :24
4BB0 53 20 4C 44 53 20 4C 44 53 20 4C 44 53 20 53 54 :23
4BC0 59 20 53 54 59 20 53 54 59 20 53 54 53 20 53 54 :7A
4BD0 53 20 53 54 53 20 53 57 49 32 24 25 27 2C 2E 22 :9E
4BE0 2F 23 2D 2B 26 2A 21 28 29 83 93 B3 A3 8C 9C BC :BC
4BF0 AC 8E 9E BE AE CE DE FE EE 9F BF AF DF FF EF 3F :F5
-------------------------------------------------------
Sum: 7E 60 9F 74 81 73 CA 3E F9 9D 48 CB 31 39 F8 33 :2B
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4C00 4E 45 47 20 20 20 20 20 20 20 20 20 43 4F 4D 20 :F9
4C10 4C 53 52 20 20 20 20 20 52 4F 52 20 41 53 52 20 :AA
4C20 4C 53 4C 20 52 4F 4C 20 44 45 43 20 20 20 20 20 :84
4C30 49 4E 43 20 54 53 54 20 4A 4D 50 20 43 4C 52 20 :1D
4C40 50 41 47 45 50 41 47 45 4E 4F 50 20 53 59 4E 43 :84
4C50 20 20 20 20 20 20 20 20 4C 42 52 41 4C 42 53 52 :54
4C60 20 20 20 20 44 41 41 20 4F 52 43 43 20 20 20 20 :0D
4C70 41 4E 44 43 53 45 58 20 45 58 47 20 54 46 52 20 :36
4C80 42 52 41 20 42 52 4E 20 42 48 49 20 42 4C 53 20 :EB
4C90 42 43 43 20 42 43 53 20 42 4E 45 20 42 45 51 20 :CD
4CA0 42 56 43 20 42 56 53 20 42 50 4C 20 42 4D 49 20 :FC
4CB0 42 47 45 20 42 4C 54 20 42 47 54 20 42 4C 45 20 :E0
4CC0 4C 45 41 58 4C 45 41 59 4C 45 41 53 4C 45 41 55 :A1
4CD0 50 53 48 53 50 55 4C 53 50 53 48 55 50 55 4C 55 :08
4CE0 20 20 20 20 52 54 53 20 41 42 58 20 52 54 49 20 :A3
4CF0 43 57 41 49 4D 55 4C 20 20 20 20 20 53 57 49 20 :C5
-------------------------------------------------------
Sum: 07 49 E9 DC 30 43 54 91 33 63 60 AC 43 7E 75 BF :04
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4D00 4E 45 47 41 20 20 20 20 20 20 20 20 43 4F 4D 41 :3B
4D10 4C 53 52 41 20 20 20 20 52 4F 52 41 41 53 52 41 :0D
4D20 4C 53 4C 41 52 4F 4C 41 44 45 43 41 20 20 20 20 :E7
4D30 49 4E 43 41 54 53 54 41 20 20 20 20 43 4C 52 41 :F9
4D40 4E 45 47 42 20 20 20 20 20 20 20 20 43 4F 4D 42 :3D
4D50 4C 53 52 42 20 20 20 20 52 4F 52 42 41 53 52 42 :10
4D60 4C 53 4C 42 52 4F 4C 42 44 45 43 42 20 20 20 20 :EA
4D70 49 4E 43 42 54 53 54 42 20 20 20 20 43 4C 52 42 :FC
4D80 4E 45 47 20 20 20 20 20 20 20 20 20 43 4F 4D 20 :F9
4D90 4C 53 52 20 20 20 20 20 52 4F 52 20 41 53 52 20 :AA
4DA0 4C 53 4C 20 52 4F 4C 20 44 45 43 20 20 20 20 20 :84
4DB0 49 4E 43 20 54 53 54 20 4A 4D 50 20 43 4C 52 20 :1D
4DC0 4E 45 47 20 20 20 20 20 20 20 20 20 43 4F 4D 20 :F9
4DD0 4C 53 52 20 20 20 20 20 52 4F 52 20 41 53 52 20 :AA
4DE0 4C 53 4C 20 52 4F 4C 20 44 45 43 20 20 20 20 20 :84
4DF0 49 4E 43 20 54 53 54 20 4A 4D 50 20 43 4C 52 20 :1D
-------------------------------------------------------
Sum: BC E4 A0 0C 98 88 80 86 AC AA B4 86 9C 38 44 C9 :E3
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4E00 53 55 42 41 43 4D 50 41 53 42 43 41 53 55 42 44 :93
4E10 41 4E 44 41 42 49 54 41 4C 44 41 20 20 20 20 20 :A5
4E20 45 4F 52 41 41 44 43 41 4F 52 41 20 41 44 44 41 :3C
4E30 43 4D 50 58 42 53 52 20 4C 44 58 20 20 20 20 20 :C7
4E40 53 55 42 41 43 4D 50 41 53 42 43 41 53 55 42 44 :93
4E50 41 4E 44 41 42 49 54 41 4C 44 41 20 53 54 41 20 :2D
4E60 45 4F 52 41 41 44 43 41 4F 52 41 20 41 44 44 41 :3C
4E70 43 4D 50 58 4A 53 52 20 4C 44 58 20 53 54 58 20 :6E
4E80 53 55 42 41 43 4D 50 41 53 42 43 41 53 55 42 44 :93
4E90 41 4E 44 41 42 49 54 41 4C 44 41 20 53 54 41 20 :2D
4EA0 45 4F 52 41 41 44 43 41 4F 52 41 20 41 44 44 41 :3C
4EB0 43 4D 50 58 4A 53 52 20 4C 44 58 20 53 54 58 20 :6E
4EC0 53 55 42 41 43 4D 50 41 53 42 43 41 53 55 42 44 :93
4ED0 41 4E 44 41 42 49 54 41 4C 44 41 20 53 54 41 20 :2D
4EE0 45 4F 52 41 41 44 43 41 4F 52 41 20 41 44 44 41 :3C
4EF0 43 4D 50 58 4A 53 52 20 4C 44 58 20 53 54 58 20 :6E
-------------------------------------------------------
Sum: 70 FC A0 6C 38 B4 E4 8C E8 70 74 84 82 9C 23 14 :79
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4F00 53 55 42 42 43 4D 50 42 53 42 43 42 41 44 44 44 :75
4F10 41 4E 44 42 42 49 54 42 4C 44 42 20 20 20 20 20 :A8
4F20 45 4F 52 42 41 44 43 42 4F 52 42 20 41 44 44 42 :40
4F30 4C 44 44 20 20 20 20 20 4C 44 55 20 20 20 20 20 :F9
4F40 53 55 42 42 43 4D 50 42 53 42 43 42 41 44 44 44 :75
4F50 41 4E 44 42 42 49 54 42 4C 44 42 20 53 54 42 20 :31
4F60 45 4F 52 42 41 44 43 42 4F 52 42 20 41 44 44 42 :40
4F70 4C 44 44 20 53 54 44 20 4C 44 55 20 53 54 55 20 :20
4F80 53 55 42 42 43 4D 50 42 53 42 43 42 41 44 44 44 :75
4F90 41 4E 44 42 42 49 54 42 4C 44 42 20 53 54 42 20 :31
4FA0 45 4F 52 42 41 44 43 42 4F 52 42 20 41 44 44 42 :40
4FB0 4C 44 44 20 53 54 44 20 4C 44 55 20 53 54 55 20 :20
4FC0 53 55 42 42 43 4D 50 42 53 42 43 42 41 44 44 44 :75
4FD0 41 4E 44 42 42 49 54 42 4C 44 42 20 53 54 42 20 :31
4FE0 45 4F 52 42 41 44 43 42 4F 52 42 20 41 44 44 42 :40
4FF0 4C 44 44 20 53 54 44 20 4C 44 55 20 53 54 55 20 :20
-------------------------------------------------------
Sum: 94 D8 70 98 31 84 88 98 E8 70 70 88 3A 58 25 18 :68
```

スクリーン・アセンブラ BASICプログラム

```
100 OPEN "O",1,"LPTO:S":GOTO 500
110 POKE PP,INT(PD/256):POKE PP+1,INT(PD-256*(INT(PD/256))):RETURN
500 DEFINT A-N:DEFSTR S-W
600 DEFUSR0=&H400A:DEFUSR1=&H4029:DEFUSR2=&H4050:DEFUSR3=&H409C:DEFUSR4=&H40F2:D
EFUSR5=&H416D
700 DEF FNS2(P)=RIGHT$("00"+HEX$(P),2)
710 DEF FNSP2(P)=RIGHT$("00"+HEX$(PEEK(P)),2)
720 DEF FNS4(P)=RIGHT$("000"+HEX$(P),4)
730 DEF FNSD(D)=MID$(STR$(D),2)
740 DEF FNSRG(U$,K)=MID$(U$,K,2+(42=ASC(MID$(U$,K+1))))
750 DEF FNPL(S)=VAL("&H"+LEFT$(RIGHT$("000"+S,4),2))
760 DEF FNPR(S)=VAL("&H"+RIGHT$("0"+S,2))
770 DEF FNRE(P)=256*(PEEK(P))+PEEK(P+1)
780 VPNT="XYUS":VSTACK="CCA*B*DPX*Y*USPC":VEXG="D*X*Y*U*S*PC****A*B*CCDP"
800 ON KEY(1) GOSUB 9999:KEY(1) ON
810 DIM AX(15),AY(15),G(12)
898 ' * * · · SELECT SWITCH · · * *
900 COLOR6:ON ERROR GOTO 9094
910 WIDTH40,25:F=0:LOCATE3,3:PRINT"1.Monitor":PRINT:PRINT"     2.Transferor":PRIN
T:PRINT"     3.Clearer":LOCATE8,11
920 COLOR7:PRINT"4.Dis-Assembler":PRINT:PRINT"          5.Assembler":LOCATE5,18
930 COLOR6:PRINT"6.C-Saver":PRINT:PRINT"   7.C-Loader":PRINT:PRINT" 8.Skipper":C
OLOR7
```

```
932 LINE@(128,112)-(143,113),PSET,6,BF:GET@A(128,112)-(143,113),G,G
940 S=INKEY$:C=ASC(S)
950 IF (C<48)+(56<C) THEN  940
960 ON C-47 GOTO 5800,1000,5200,5400,2000,3000
970 ON ERROR GOTO 9500
980 ON C-53 GOTO 5000,5020,5040
998 ' * * · · MONITOR · · * *
1000 WIDTH40:HX=0:HY=0:GOSUB 1990
1010 COLOR5:INPUT"Monitor";T:P=VAL("&H"+T):WIDTH80
1020 EP=-(0<>INSTR(T,"P")):EV=15+8*(0<>INSTR(T,"X"))+12*(0<>INSTR(T,"Y"))+14*(0<
>INSTR(T,"Z")):EV=ABS(EV)
1030 T="    +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  ;Sum"
1050 LOCATE0,1:COLOR4:PRINT T
1060 FOR J=0 TO EV:PM=P+16*J:T=FNS4(PM)+" ":IF F THEN J=EV:NEXT:GOTO 900
1070 TS=" ":FOR I=0 TO 15:D=PEEK(PM+I):T=T+" "+FNS2(D):AY(J)=AY(J)+D:AX(I)=AX(I)
+D:IF (D<32)+(127=D) THEN 1074
1072 TS=TS+CHR$(D):NEXT:GOTO 1100
1074 TS=TS+".":NEXT
1100 IF (J MOD 4=0) THEN PRINT
1110 COLOR5:PRINT LEFT$(T,4):COLOR7:LOCATE5,CSRLIN-1:PRINT MID$(T,6):COLOR4:LOCA
TE55,CSRLIN-1:PRINT ";"+FNS2(AY(J)):LOCATE60,CSRLIN-1:COLOR6:PRINT TS:NEXT:IF EV
<15 THEN 1150
1120 T=" Sum ":FOR I=0 TO 15:T=T+" "+FNS2(AX(I)):ZS=ZS+AX(I):NEXT:T=T+" <"+FNS4(
ZS)+">"
1130 COLOR4:PRINT:PRINT T:COLOR7
1140 IF EP THEN HARDC
1150 EM=1:ON ERROR GOTO 9117
1160 GOSUB 1800:COLOR7
1170 C=ASC(INKEY$)
1180 GOSUB 1800:IF (27<C)*(C<32) THEN GOSUB 1850:GOTO 1160
1190 IF (47<C)*(C<58) THEN 1210 ELSE IF (64<C)*(C<71) THEN 1220
1200 IF 12=C THEN 1000 ELSE IF 27=C THEN 1300 ELSE 1160
1210 D=C-48:GOTO 1230
1220 D=C-55
1230 EPRTY=HX MOD 2:PM=P+HY*16+HX¥2:DO=PEEK(PM):D=EPRTY*((DO¥16)*16+D)+(1-EPRTY)
*(D*16+(DO MOD 16))
1240 POKE PM,D:GOSUB 1900:HY=HY-(31=HX):C=28:GOSUB 1850:GOTO 1160
1300 EM=2:ON ERROR GOTO 9132
1310 GOSUB 1950
1320 C=ASC(INKEY$):GOSUB 1950:IF (27<C)*(C<32) THEN GOSUB 1850:GOTO 1310
1330 IF 12=C THEN 1000 ELSE IF 27=C THEN 1150
1340 PM=P+16*HY+HX¥2:POKE PM,C:GOSUB 1900:HY=HY-(29<HX):C=28:GOSUB 1850:GOTO 131
0
1800 LINE@(48+8*(HX+HX¥2),24+8*(HY+HY¥4))-(55+8*(HX+HX¥2),31+8*(HY+HY¥4)),XOR,6,
BF:RETURN
1850 HX=(HX+32-EM*(28=C)+EM*(29=C))MOD 32:HY=(HY+EV+1-(31=C)+(30=C))MOD (EV+1):R
ETURN
1900 D=PEEK(PM):LOCATE 6+3*(HX¥2),3+HY+HY¥4:PRINT FNS2(D)
1910 LOCATE 61+HX¥2,3+HY+HY¥4:IF (D<32)+(127=D) THEN S="·":GOTO 1920 ELSE S=CHR$
(D)
1920 COLOR6:PRINT S:RETURN
1950 LINE@(488+8*(HX¥2),24+8*(HY+HY¥4))-(495+8*(HX¥2),31+8*(HY+HY¥4)),XOR,7,BF:R
ETURN
1990 FOR I=0 TO 15:AX(I)=0:AY(I)=0:NEXT:AZ=0:RETURN
1998 ' * * · · DIS-ASSEMBLER · · * *
2000 CLS:ON ERROR GOTO 9200
2100 COLOR5:INPUT"Address";T:DUM=ASC(T):PC=VAL("&H"+T):EL=0:EP=INSTR(T,"P"):IF E
P THEN PRINT# 1," "
2200 FRG=0:FLB=0:FS=0:FIND=0:IF F THEN 900
2210 W=" ":WA=" ":WBR=" ":PCO=PC:PD=PC:GOSUB 7000
2220 IF D=16 THEN GOSUB 7300 ELSE IF D=17 THEN GOSUB 7400 ELSE GOSUB 7200
2300 DUM=D:DO=D:A=USRO(DUM)
2320 AF=A¥4:AL=A MOD 4:IF AL=0 THEN ERROR 90
2330 IF AF=7 THEN GOSUB 6000
2335 IF FIND THEN ERROR 91
2340 IF FLB=0 THEN 2370
2350 AL=3
2370 IF 1=AL THEN 2440
2400 FOR PI=PC+1 TO PC+AL-1:W=W+FNSP2(PI):NEXT:W=MID$(W,2)
2440 D=DO:ON AF+1 GOSUB 8100,8000,8300,8020,8020,8040,8060
2445 D=DO
2450 IF(D=14)+(D=126)+(D=157)+(D=189)THEN 8200 ELSEIF(D=110)+(D=173)THEN 8220
2500 COLOR5:PRINT FNS4(PCO):COLOR7:LOCATE14,CSRLIN-1:PRINT U:SYMBOL(304,8*(CSRLI
N-1)),WA,2,1,7-3*FRG,0,XOR
2520 SYMBOL(96,8*CSRLIN-8),SCODE+W,1,1,7,0,XOR:IF 0=EP THEN 2600
2540 PRINT# 1,FNS4(PCO),SCODE+W,U+WA,WBR
2600 PC=PC+AL:EL=EL+1:IF EL<20 THEN 2200 ELSE 2100
2998 ' * * · · SCREEN EDITOR · · * *
3000 CLS:HX=16:HY=2
```

スクリーン・アセンブラ BASICプログラム

```
3050 LINE@(0,15)-(639,15),PSET,6:LINE@(0,176)-(639,176),PSET,6
3060 LINE@(143,12)-(144,14),PSET,6,BF:LINE@(255,12)-(256,14),PSET,6,BF
3100 SYMBOL(8,0),"FM-8 SCREEN ASSEMBLER",1,1,6
3120 GOSUB 3900:LOCATE0,10:INPUT"First Line No.";EL:IF (EL<1)+(EL>81) THEN 3120
3130 GOSUB 3900:FOR I=0 TO 19:SYMBOL(0,8*I+16),FNSD(EL+I),1,1:NEXT
3140 FOR I=0 TO 19:POKE &H3C40,EL+I:UL=SPACE$(32):UL=USR5(UL)
3150 LOCATE 9,I+2:COLOR4:PRINT LEFT$(UL,7):LOCATE 16,I+2:COLOR7:PRINT MID$(UL,8,
24):NEXT
3180 EP=0:EB=0:ON ERROR GOTO 9320:GOSUB 3800
3200 S=INKEY$:C=ASC(S)
3210 GOSUB 3800:IF 31<C THEN 3320ELSE IF 27<C THEN 3600
3218 ON  C GOTO 4000,3290,3250,5600,3180,3180,3180,3240,5500,3180,3230,3120,3220
,3180,3180,3300
3219 GOTO 3180
3220 HX=16:HY=HY+1:GOTO 3620
3230 HX=16:GOTO 3180
3240 HX=9:GOTO 3180
3250 EXEC &H4156:GOTO 3120
3290 EB=1
3300 EP=1:GOTO 4000
3320 C=C+31*((91=C)+(93=C))+59*(94=C)+56*(92=C):S=CHR$(C)
3390 LINE@(16*HX,8*HY)-(16*HX+15,8*HY+7),PSET,0,BF
3400 SYMBOL(16*HX,8*HY),S,2,1,7+(HX<16)*3,0,PSET
3410 POKE &H5000+HX-9+31*(EL+HY-3),C:C=28
3600 HX=HX-(28=C)+(29=C):HY=HY-(31=C)+(30=C)
3620 HX=(HX+22) MOD 31+9:HY=(HY+18) MOD 20+2:GOTO 3180
3800 PUT@A(16*HX,8*HY+6)-(16*HX+15,8*HY+7),G,XOR:RETURN
3900 LINE@(0,16)-(639,175),PSET,0,BF:RETURN
3998 ' * * ・ ・ ASSEMBLER ・ ・ * *
4000 LINE(0,177)-(639,199),PSET,0,BF:LINE@(24,16)-(143,175),PSET,0,BF:IF O=EP TH
EN DF=150:DR=150:GOTO 4040
4010 LOCATE0,23:INPUT"LIST";T:DF=VAL(LEFT$(" "+T,INSTR(T,"-"))):DF=-1*(O=DF)+DF:
DR=VAL(MID$(T+" ",INSTR(T,"-")+1)):DR=-100*(O=DR)+DR MOD 256:IF EB THEN WIDTH80:
CLOSE 1:OPEN"O",1,"SCRN:"
4020 PRINT#1,CHR$(10)+CHR$(10)+"FM-8   6809-self Assembler Ver.1.0   DATE"+STR$(
DATE)+CHR$(10)
4040 PO=&H5C20:POO=PO:POKE &H3C91,EL-1:POKE &H3C92,EL+20:POKE &H3C93,DF-1:POKE &
H3C94,DR+1
4050 POKE &H3C90,0:POKE &H3C8F,0:EXEC &H3A40
4060 EEN=PEEK(&H3C8F):ELN=PEEK(&H3C40):IF EEN THEN 4300
4070 IF (ELN=EL-1)+(ELN=DF-1) THEN PO=FNRE(&H5C1E):GOTO 4250
4080 IF (EL-1=<ELN)*(ELN=<EL+20)+(DF-1=<ELN)*(ELN=<DR+1) THEN 4100 ELSE 4250
4100 P1=FNRE(&H5C1E):PO1=PO
4110 W=" ":WHILE PO<P1
4120 W=W+FNS2(PEEK(PO))
4130 PO=PO+1:WEND:ELOC=PO1-POO+FNRE(&H3C80)
4140 IF (EL=<ELN)*(ELN=<EL+19)*(O=EB) THEN SYMBOL(32,16+8*(ELN-EL)),W,1,1,6,0,PS
ET
4150 IF (DF=<ELN)*(ELN=<DR) THEN 4170 ELSE 4250
4170 PRINT# 1,"   "+STR$(ELN)+".",FNS4(ELOC)+" <"+FNS4(PO1)+">",W,LEFT$(USR5(UL),
31)
4250 IF ELN=100 THEN 4270
4260 EXEC &H3AEA:GOTO 4060
4270 WUN="Undefined Labels"+STR$(PEEK(&H3C90))
4280 WO="Origin="+FNSP2(&H3C80)+FNSP2(&H3C81)+"   CONT="+FNSP2(&H3C88)+FNSP2(&H3C
89):IF EP THEN PRINT# 1,CHR$(10)+CHR$(10)+WO:PRINT# 1,WUN
4290 IF O=EB THEN SYMBOL(400,184),WO,1,1,7,0,PSET:SYMBOL(400,192),WUN,1,1,6,0,PS
ET:GOTO 3180
4295 IF F THEN CLOSE 1:OPEN"O",1,"SCRN:":GOTO 900ELSE 4295
4300 ON EEN GOTO4310,4320,4330,4340,4350,4360,4370,4380,4390,4400,4410,4420,4430
4310 WE="Addr Err":GOTO4500
4320 WE="Long Mnem":GOTO4500
4330 WE="Long Addr":GOTO4500
4340 WE="Long Lbl":GOTO4500
4350 WE="Strct Err":GOTO4500
4360 WE="Syntx Err":GOTO4500
4370 WE="Symb Ovfw":GOTO4500
4380 WE="Symb Err":GOTO4500
4390 WE="Obj Ovfw":GOTO4500
4400 WE="Remot Lbl":GOTO4500
4410 WE="Lbl Ovfw":GOTO4500
4420 WE="Lbl Err":GOTO4500
4430 WE="Mnem Err"
4500 LINE@(292,184)-(571,191),PSET,2,BF
4510 SYMBOL(300,184),"Line No."+STR$(ELN)+";   "+WE,1,1,7,0,PSET:GOTO 3180
4998 ' * * ・ ・ MODULE ・ ・ * *
5000 CLS:INPUT "Save";T:SAVEM T,&H5000,&H5C1C,&H5000:GOTO 900
5020 CLS:INPUT "Load";T:DUM=ASC(T):LOADM T:GOTO 900
5040 CLS:INPUT "Skip";T:DUM=ASC(T):SKIPF T:GOTO 900
5199 'Transferor
```

```
5200 CLS:ON ERROR GOTO 9552
5210 COLOR4:PRINT"Assembled Program ?  (Y,N)":COLOR7
5220 S=INKEY$:C=ASC(S):IF C=89 THEN 5260 ELSE IF C=78 THEN 5230 ELSE 5220
5230 INPUT"from";TF:INPUT"  to";TT
5240 POKE &H3C10,FNPL(TF):POKE &H3C11,FNPR(TF)
5250 POKE &H3C12,FNPL(TT):POKE &H3C13,FNPR(TT):GOTO 5270
5260 PP=&H3C10:PD=&H5C20:GOSUB 110:PD=256*PEEK(&H5C1E)+PEEK(&H5C1F)-1:PP=&H3C12:
GOSUB 110
5270 PRINT:PRINT:PRINT:COLOR4:PRINT"Refer to ORGN ?  (Y,N)":COLOR7
5280 S=INKEY$:C=ASC(S):IF C=89 THEN 5310 ELSE IF C=78 THEN 5290 ELSE 5280
5290 INPUT"Destination";TD
5300 POKE &H3C14,FNPL(TD):POKE &H3C15,FNPR(TD):GOTO 5350
5310 POKE &H3C14,PEEK(&H3C80):POKE &H3C15,PEEK(&H3C81)
5350 FCT=0:EXEC&H4000
5360 LOCATE30,16:PRINT"O.K."
5370 IF INKEY$=CHR$(12) THEN 5380 ELSE IF F THEN 900 ELSE 5370
5380 IF FCT THEN 5400 ELSE 5200
5399 'Clearer
5400 CLS:COLOR3:PRINT"Clear":COLOR7
5410 INPUT"  from";TF:INPUT"    to";TT
5420 POKE &H3C96,FNPL(TF):POKE &H3C97,FNPR(TF)
5430 POKE &H3C98,FNPL(TT):POKE &H3C99,FNPR(TT)
5440 EXEC &H48EF:FCT=1:GOTO 5360
5499 ' Insert
5500 POKE &H3C12,&H5B:POKE &H3C13,&HFD
5510 PD=&H5000+31*(EL+HY-2):PP=&H3C10:GOSUB 110
5520 PD=PD+31:PP=&H3C14:GOSUB 110
5540 EXEC &H4000:POKE &H3C40,(EL+HY-1):EXEC &H47E0
5550 FOR I=HY-2 TO 19:POKE &H3C40,EL+I:UL=SPACE$(32)
5560 UL=USR5(UL):LOCATE 9,I+2:COLOR4:PRINTLEFT$(UL,7):LOCATE 16,I+2:COLOR7:PRINT
MID$(UL,8,24):NEXT:GOTO 3180
5599 ' Delete
5600 POKE &H3C12,&H5C:POKE &H3C13,&H1B
5610 PD=&H5000+31*(EL+HY-3):PP=&H3C14:GOSUB 110
5620 PD=PD+31:PP=&H3C10:GOSUB 110
5640 EXEC &H4000:POKE &H3C40,100:EXEC &H47E0:GOTO 5550
5998 '* * · · SUBROUTINE · · * *
5999 'Index Post-byte
6000 PD=PC+1:GOSUB 7000
6010 WR=MID$(VPNT,2*B(6)+B(5)+1,1):IF B(7)=0 THEN 6100
6020 D5=D MOD 32:POKE &H3C3B,D5
6030 WA=CHR$(PEEK(PC+2))+CHR$(PEEK(PC+3))+SPACE$(10)
6040 WA=USR4(WA)
6050 AL=AL+1+PEEK(&H3C54):IF ASC(WA)<>69 THEN 6060ELSE FIND=1:RETURN
6060 HR=INSTR(WA,"R"):IF HR THEN 6070 ELSE RETURN
6070 MID$(WA,HR)=WR:RETURN
6100 D5=D MOD 32:IF D5<16 THEN WDEC="%+"+FNSD(D5):WA="<"+WR+">"+WDEC:AL=2:RETURN
6110 D5=D5-32:WDEC="%-"+FNSD(D5):WA="<"+WR+">"+WDEC:AL=2:RETURN
6998 '
7000 PP=&H3C08:GOSUB 110:EXEC &H40E8
7010 FOR I=0 TO 7:B(I)=PEEK(&H3C00+I):NEXT
7020 D=PEEK(PD):DH=D¥16:DL=D MOD 16:RETURN
7199 'PAGE1
7200 POKE &H3C36,D:U="TFR *"
7230 U=LEFT$(USR1(U),4)
7240 IF (D=22)OR(D=23) THEN 7250 ELSE 7260
7250 FLB=1
7260 SCODE=FNSP2(PC):RETURN
7299 'PAGE2
7300 PC=PC+1:PD=PC:GOSUB 7000
7310 POKE &H3C36,D:U="LDY *"
7330 U=LEFT$(USR2(U),4)
7340 IF ASC(MID$(U,2,1))<>66 THEN 7360
7350 FLB=1
7360 SCODE="10"+FNSP2(PC):RETURN
7399 'PAGE3
7400 PC=PC+1:PD=PC:GOSUB 7000
7410 POKE &H3C36,D:U="     *"
7430 U=LEFT$(USR3(U),4)
7440 SCODE="11"+FNS2(PC):RETURN
7999 'Addressing String
8000 RETURN
8020 WA="#"+W:RETURN
8040 WA="<P/"+W:RETURN
8060 WA=W:RETURN
8099 'Paint on JMP
8100 WA=W:IF FLB THEN 8140
8120 BR=VAL("&H"+W):IF BR<&H80 THEN 8170
8130 BR=BR-256:GOTO 8170
8140 ZBR=VAL("&H"+W):IF ZBR<&H8000 THEN 8160
```

スクリーン・アセンブラ　BASICプログラム

```
8150 ZBR=ZBR-65536!
8160 BR=ZBR
8170 PJ=PC+AL+BR:WBR="0000"+HEX$((PJ<2^16)*(-PJ)+(PJ>=2^16)*(2^16-PJ))
8180 LINE@(408,8*CSRLIN)-(488,8*CSRLIN+7),XOR,6,BF:WBR=RIGHT$(WBR,4)
8190 SYMBOL(416,8*CSRLIN),WBR,2,1,1,0,PSET:RETURN
8200 LINE@(304,8*CSRLIN)-(384,8*CSRLIN+7),XOR,6,BF:GOTO 2500
8220 LINE@(304,8*CSRLIN-1)-(432,8*CSRLIN+7),PSET,6,B:GOTO 2500
8299 'Register Addressing
8300 IF (D=30)+(D=31) THEN 8400 ELSE IF (D=53)+(D=55) THEN 8350
8310 FS=1
8349 'Stack
8350 PD=PC+1:GOSUB 7000:WA=" "
8360 FOR I=7*FS TO 7*(1-FS) STEP 1-2*FS:IF B(I) THEN 8370 ELSE 8380
8370 WA=WA+FNSRG(VSTACK,2*I+1)+","
8380 NEXT:FRG=1
8390 WA=LEFT$(WA,LEN(WA)-1):RETURN
8399 'EXG
8400 PD=PC+1:GOSUB 7000
8410 WA=" "+FNSRG(VEXG,2*DH+1)+"-"+FNSRG(VEXG,2*DL+1)
8420 FRG=1:RETURN
8998 ' * * ・ ・ Error Trap・ ・ * *
9000 PRINT ERR,ERL:STOP
9094 IF ERL=940 THEN RESUME 940 ELSE 9000
9117 IF F THEN RESUME 900 ELSE IF ERL=1170 THEN RESUME 1170 ELSE 9000
9132 IF F THEN RESUME 900 ELSE IF ERL=1320 THEN RESUME 1320 ELSE 9000
9200 IF ERL=2100 THEN EL=0:RESUME 2200
9210 IF ERR=90 THEN WE=SCODE+"Code Error" ELSE IF ERR=91 THEN WE="Index Error" E
LSE 9000
9220 LINE@(520,8*CSRLIN-1)-(639,8*CSRLIN+7),PSET,2,BF
9230 SYMBOL(528,8*CSRLIN),WE,1,1,7,0,PSET:RESUME 2100
9320 IF F THEN RESUME 900 ELSE IF ERL=3200 THEN RESUME 3200 ELSE 9000
9500 IF F THEN RESUME 900 ELSE STOP
9552 IF ERL=5220 THEN RESUME 5220ELSE IF ERL=5280 THEN RESUME5280ELSE 9000
9998 '
9999 F=1:RETURN
```



FM-8 6809 エディタ・アセンブラ

■馬場二郎

このアセンブラの特徴は，

❶ 機能は必要最小限とする．
❷ すべてマシン語である．
❸ 256バイト単位($××○○)でポジション・インディペンデントである．
❹ 他機種への移殖，メモリ・マップの変更が，容易である．

です．

FM-8は，他のパーソナル・コンピュータと違い，ほとんどの入出力をサブCPUで行なっていますが，そのBIOSについてはあまり説明されていません．

ここでは，私なりに次の3つの基本的ルーチンを，BIOSエントリ・ルーチンを介して実現しています．

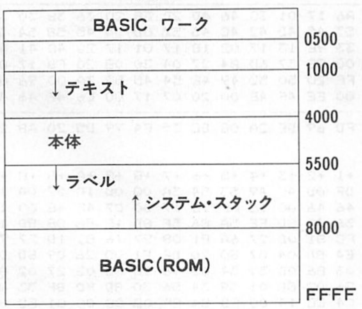
❶ キーボードの入力を(A)に入れて戻る(押されないときは0)．
❷ (A)のキャラクタ・コードを画面へ出力．
❸ (A)のキャラクタ・コードをプリンタに出力．

これらのルーチンが使えるものであれば移殖は容易にできます．

メモリ・マップ

現在は図1のようなマッピングになっていますが，本体はポジション・インディペンデントによりRAM中のどこへ置いてもかまいません．テキストとラベル領域については，本体バッファ中の＄4038および＄4036の内容を，使用したい領域の先頭アドレスに換えるだけでできます．

図1



使い方

残念ながらスクリーン・エディタおよびその関連するキーは使えません．使えるのは BS のみです．

MONコマンドによりBASICのモニタ・モードにし，G4000 CR を行なうと，システムのリポートとともに″>－″を表示して，コマンドの入力待ちになります．

コマンドを表1に示します．

表1　コマンド一覧

コマンド	機能
1	ラベルの登録
2	アセンブル実行(リストSWオンならプリントも行なう)
B	BASIC MONモードへ
D××××〔, ××××〕	テキストの削除
I××××	テキストの追加
L〔××××〕	テキスト表示
M××××	××××行を削除し，その場所からテキストの入力を行なう
P	プリンタ・スイッチのON/OFF
R	システム・リポート
S	テキスト・クリア
T	アセンブル時にディスプレイに表示しない(エラーのみ出力される)

××××は16進(4桁MAX)．〔 〕は省略可．

I0 CR を行なうと改行して″－″だけの表示となり，テキストを入力できます．フォーマットは，

```
␣LABEL : ␣LDA␣$1, X␣ ; コメント CR
         ↑       ↑      └コメントの前に必須
         │       └オペランドの前に必須
         └ラベルの後に必須        ␣(スペース)はいくつでも良い．
                                  例．ABX CR
```

です．

CR 入力後，フォーマット・チェックとラベルを除くニモニックのチェックを行ないます．エラーが出るとテキストは追加されないので，もう一度入力してください．何も入力せず CR で再び，コマンド・モードに戻ります．

L CR で，入力したテキストを表示します．途中スペース・バーを押すと，リストが一旦止まります．続けたければ，再度スペース・バーを，コマンド・モードに戻るときには，他のキーを押します(パス2実行中でも同様です)．

アセンブルを行なうときには必ずパス1でラベルを登録してください．『1』が不要なのはラベルのアドレスに変更がないときに限ります．『1』でラベル名が出てくるのは重複です．PGMERRは本来出るはずはありませんが，その場合にはテキストの内容が何らかの影響で，変わっていることが考えられます．

パス2でアセンブル実行中，オブジェクトに″－－″が表示されるのは，オペランドに使っているラベルが登録されていないもので，PGMERRは相対アドレスがオーバーしていることを示します．どちらも正常には動作しません．

テキストのSAVE，LOADは本アセンブラでは管理しません．″B″コマンドでBASICに戻りSAVEM，LOADMで行なってください．

ニモニック

命令はほとんどモトローラ仕様になっていると思いますが，擬似命令やラベルの使用はかなり使い方に制限があります．

▶擬似命令

FCB…1バイト定数．

EQU…ラベルの登録．

FDB…2バイト定数．

RMB…領域の割付け．

FCC…文字データ．〝/〟スラッシュで囲む．

ORG…オブジェクトの先頭アドレス．

＝………EQUと同じ．

▶定　　数　10進…符号なしまたは〝−〟

16進…〝＄〟

文字…〝'〟で1バイトのみ修飾

▶そ の 他　ANDCC→ANDC（4バイトに統一のため）．

BLO→BCS．BHS→BCC．

特殊レジスタは，ダイレクト・ページ(DP)，コンディション・コード(CC)，プログラム・カウンタ(PC)です．

ラベル，定数の演算

⊕⊖の演算のみ許します（ラベル演算には先にラベルがあることが必要：1＋LABELは許されない）．

FCB…（10進，16進，文字定数の演算）

EQU…（　　　〃　　　），＊

RMB…（　　　〃　　　）

FDB…（ラベル，16進，10進，文字定数の演算）

＊：現在のアドレス

オフセット

ラベルが使えると同時に演算も行なえるため，次のようにモードを決めます．演算中1つでも10進以外があれば，すべてHEX形式のオフセットとみなします．

HEX形式…すべて16ビット・オフセット，8ビットにするには〝<〟で強制的に行なう．5ビット・オフセットの指定はできない．

10進形式…符号付きと見て，自動的に5ビット，8ビット，16ビット・モードを決める．

おわりに

実際使っていての問題点は，擬似命令の不足や，ラベル使用上の制限よりむしろ，オブジェクトにして2K以上になったときのアセンブルの速度と作成したマシン語のテストの方法にあるようです．アセンブル速度は表示に最も時間がかかっているようなので，不要な所は表示せずに行なうように改良したほうが良いようです．テストの道具としては現在BASICのモニタがありますが，割り込みを禁止してブランチするなど，とても実用的ではありません．FIRQのジャンプ・ベクター＄FFF6は書換可能ですから，これを利用したモニタを作成することが望まれます．

参考までに私の行なったデバッグ方法は，まずアセンブルしたらすぐオブジェクトとテキストをテープにSAVEします．実行すると，たいてい制御不可能になります．このとき **STOP** を押しながら **RESET** をかけると，運が良ければBASICモードに戻ります（ダメならテープから戻す）．

リストを調べ，ハンド・アセンブルでオブジェクトを直します（テキストを直して再度アセンブルしない点がミソ）．2バイトの所へ3バイト命令を入れたいときは，JMPやJSRで他の所で処理をして戻してやります．こうして同じことを繰り返し，OKになったらテキストを直して正式なアセンブル・リストを得ます．長いものなら何回もアセンブルするより効率的です．

◘参考文献

1) 〝8080対話型2パス・アセンブラ〟I/O，1979.11

2) 〝Z-80/8080アセンブラ〟インターフェース，1980.12

3) 〝FM-8ソフトウェア概説下〟インターフェース，1981.11

4) 〝6809ハンドブック〟アスキー出版

アセンブラ ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4000 1C 00 16 01 16 00 00 00 00 00 00 00 00 00 00 00 :49
4010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
4020 00 00 00 00 00 00 00 00 00 00 00 2A 0D 3B 2A 20 :BC
4030 44 45 42 00 00 00 00 55 00 10 00 00 00 00 00 35 :65
4040 2E 31 35 20 20 20 20 20 20 20 2A 0D 3B 2A 2A 2A :64
4050 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A :A0
4060 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 0D 3B 0D :77
4070 3B D3 B0 C4 DE C1 AA AF B8 20 D3 DA 0D 20 4F 52 :CD
4080 00 20 24 34 41 46 31 0D 20 4C 42 53 52 20 4C 34 :30
4090 37 33 30 0D 20 4C 44 41 20 34 2C 55 0D 3B 0D 20 :E2
40A0 4F 52 47 20 24 34 42 31 34 0D 20 4C 42 53 52 20 :87
40B0 4C 34 37 33 30 0D 20 4C 44 41 20 34 2C 55 0D 3B :35
40C0 0D 20 4F 52 47 20 24 34 37 33 30 0D 4F 42 4A 41 :50
40D0 3D 24 34 41 35 44 0D 45 52 52 33 3D 24 34 31 30 :6E
40E0 30 0D 4C 34 37 33 30 3A 4C 44 41 20 2C 55 0D 20 :30
40F0 41 4E 44 41 20 23 31 0D 20 42 45 51 20 45 52 52 :96
---------------------------------------------------------
Sum: AA 15 76 D5 F0 C2 87 03 D9 7D E8 48 35 CF 9A 9A :04


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4100 16 02 05 16 03 6C 16 01 3F 16 02 2F 16 03 BC 16 :2A
4110 03 75 16 02 CD 16 03 24 16 03 00 30 8D FE E6 1F :73
4120 10 1F 8B 0F 05 0F 1D 0F 10 0F 1F 20 70 0F 05 0F :FA
4130 1D 0F 1F 86 0D 17 01 10 86 3E 17 01 0B 17 00 E0 :E4
4140 A6 A4 81 4C 10 27 01 CD 81 44 10 27 05 23 81 49 :0A
4150 10 27 03 C8 81 32 10 27 06 29 81 31 10 27 06 20 :2A
4160 81 50 27 23 81 42 27 16 81 53 27 15 81 52 27 2D :52
4170 81 54 10 27 00 1D 81 4D 10 27 04 FE 20 AF 7E AF :2C
4180 A2 9E 39 6F 84 20 A6 96 0F 27 04 0F 0F 20 0E 03 :51
4190 0F 20 0A 0D 10 27 04 0F 10 20 02 03 10 86 0D 17 :7F
41A0 00 A6 17 01 3C 46 4D 2D 38 20 36 38 30 39 20 41 :4A
41B0 53 53 45 4D 42 4C 45 52 0D 54 45 58 54 3A 00 17 :00
41C0 03 33 9E 13 17 02 1B 17 01 17 20 4C 41 42 45 4C :CA
41D0 3A 00 9E 37 6D 84 27 04 30 08 20 F8 17 02 03 17 :AE
41E0 00 FF 20 50 52 49 4E 54 45 52 3A 00 96 0F 27 08 :51
41F0 17 00 EE 4F 4E 00 20 07 17 00 E6 4F 46 46 00 17 :B8
---------------------------------------------------------
Sum: 56 FD 69 BE 2A 08 DC 35 F4 79 D5 20 AB 24 7D 5D :C8


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4200 00 DF 0D 4C 49 53 54 3A 00 0D 10 27 0A 17 00 D1 :98
4210 4F 46 46 00 16 FF 16 17 00 C7 4F 4E 00 16 FF 0D :A3
4220 34 56 30 8D FE 5A 86 5F 8D 1E 86 08 8D 1A 8D 54 :45
4230 27 FC 81 0D 27 6A 81 08 27 76 81 1D 27 72 81 20 :40
4240 25 E4 8D 04 A7 80 20 DE 81 0D 26 09 8D 07 86 0A :A0
4250 8D 03 86 0D 39 34 02 96 1D 35 02 27 02 8D 64 0D :A3
4260 1F 26 02 8D 01 39 34 56 30 8D FD BF 33 8D FD C4 :92
4270 A7 C4 CC 14 00 ED 84 EF 02 CC 00 01 ED 04 AD FF :17
4280 FB FA 35 D6 34 30 31 8D FD A9 6F A4 30 8D FD 9B :30
4290 CC 15 00 ED 84 10 AF 02 AD FF FB FA A6 A4 35 B0 :E3
42A0 86 20 8D A4 86 0D A7 80 8D 9E 31 8D FD D2 35 D6 :54
42B0 6D 1F 27 0C 86 20 8D 90 86 08 8D 8C 8D 8A 6F 82 :31
42C0 16 FF 63 34 56 30 8D FD 62 33 8D FD 67 A7 C4 CC :79
42D0 0E 00 ED 84 EF 02 CC 00 01 ED 04 AD FF FB FA 35 :04
42E0 D6 AE E1 A6 80 27 05 17 FF 5E 20 F7 1F 15 8D F1 :F4
42F0 43 4D 44 45 52 52 0D 00 16 FE 32 8D E4 46 4D 54 :68
---------------------------------------------------------
Sum: 19 90 43 AE 40 08 CA 24 B9 CD 96 6F 36 68 0F 15 :1D


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4300 45 52 52 0D 00 86 01 39 8D D7 50 47 4D 45 52 52 :E7
```

```
4310 0D 00 86 01 39 A6 21 81 0D 27 0E 17 01 67 26 CE :CA
4320 8C 00 00 27 07 30 1F 20 03 8E 00 00 9F 11 17 01 :82
4330 CD 26 BB 86 0D 17 FF 10 10 9E 15 A6 A4 10 27 FD :A8
4340 EC 8D 70 17 03 A5 30 01 17 00 97 86 20 17 FE F8 :3A
4350 17 04 36 A6 A4 81 3B 27 49 C6 06 17 00 A7 A6 A0 :97
4360 81 3A 27 06 81 3D 27 33 86 20 17 FE DB C6 04 17 :77
4370 00 93 81 3B 10 27 00 EE 17 FE CD 31 21 81 0D 27 :5D
4380 2D C6 0E A6 A4 81 2F 27 19 17 00 79 31 21 81 0D :AB
4390 27 09 34 02 86 20 17 FE AF 35 02 17 FE AA 81 0D :54
43A0 27 0C C6 28 17 00 22 31 21 86 0D 17 FE 9A 17 03 :08
43B0 DB 20 88 17 FE CE 81 20 26 0E 17 FE C7 27 FB 81 :BA
43C0 20 27 05 35 06 16 FD 65 39 A6 A4 81 0D 27 12 17 :60
43D0 FE 76 31 21 5A 26 F2 A6 A4 81 0D 27 04 31 21 20 :AD
43E0 F6 39 34 04 1F 10 17 00 53 96 0D 17 FE 5A 96 0E :B6
43F0 17 FE 55 1F 98 17 00 44 96 0D 17 FE 4B 96 0E 17 :3A
-----------------------------------------------------------
Sum: B0 A5 30 19 DB CF C1 F8 7F B8 EF 32 FB A6 56 EE :3E
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4400 FE 46 35 04 39 A6 A4 81 27 27 20 17 00 64 27 0B :9C
4410 17 FE 35 31 21 5A 2A ED 5C 20 EA 5D 27 0C 34 02 :39
4420 86 20 17 FE 23 35 02 5A 20 F1 39 17 FE 1A 5A 2A :6C
4430 01 5C 31 21 A6 A4 81 0D 27 E1 20 D4 34 06 1F 89 :65
4440 84 0F 81 0A 24 14 8A 30 97 0E 54 54 54 54 C4 0F :D8
4450 C1 0A 24 0C CA 30 D7 0D 35 86 80 09 8A 40 20 E8 :EF
4460 C0 09 CA 40 20 F0 34 02 C6 10 17 FF AE 35 02 16 :00
4470 FF 30 81 20 27 0E 81 0D 27 0A 81 3A 27 06 81 3D :6A
4480 27 02 81 3B 39 34 20 31 21 8E 00 00 86 01 97 0C :7C
4490 A6 A0 80 30 25 54 81 0A 25 1C 80 07 81 10 24 4A :C1
44A0 20 14 A6 A0 8D 25 27 47 80 30 25 3E 81 0A 25 06 :63
44B0 80 07 81 10 24 34 34 06 1F 10 58 49 58 49 58 49 :BC
44C0 58 49 1F 01 35 06 1E 89 3A 20 D7 81 3B 27 1A 81 :52
44D0 20 27 16 81 2C 27 12 81 5D 27 0E 81 3A 27 0A 81 :C3
44E0 0D 27 06 81 2B 27 02 81 2D 39 35 20 86 01 39 35 :40
44F0 06 31 3F 4F 39 9E 39 A6 80 26 FC 9F 13 39 8E 00 :96
-----------------------------------------------------------
Sum: 98 97 44 37 2C EE CE DA AC 57 E2 44 FA 4B 5E E6 :1E
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4500 00 DE 39 9C 11 27 0E A6 C0 27 0E 81 0D 27 02 20 :6B
4510 F6 30 01 20 EE DF 15 4F 39 86 01 39 17 FF 66 26 :13
4520 3F 9F 11 17 00 78 26 FB 17 02 5B 26 F6 17 FF C5 :0A
4530 17 FF CB 26 2B 17 00 2B DC 15 1F 01 DC 13 93 15 :1C
4540 1F 02 DC 15 D3 17 1F 03 17 00 2C 30 8D FB 71 10 :9A
4550 9E 17 DE 15 17 00 20 DC 11 C3 00 01 DD 11 20 C3 :61
4560 16 FD 8B 30 8D FB 59 CE 00 01 A6 80 81 0D 27 04 :5D
4570 33 41 20 F6 DF 17 39 DF 19 9C 19 24 18 10 9F 19 :6A
4580 1F 10 D3 19 1F 01 1F 30 D3 19 1F 03 A6 82 A7 C2 :29
4590 31 3F 26 F8 39 A6 80 A7 C0 31 3F 26 F8 39 30 8D :D8
45A0 FB 1E 17 FC 7B 17 01 38 81 0D 10 27 00 A7 81 3B :1F
45B0 10 27 00 8C 34 20 A6 A0 17 FE B7 26 F9 35 20 81 :1E
45C0 3A 27 0A 81 3D 27 06 86 20 A7 80 20 2B C6 07 A6 :E1
45D0 A4 A7 80 81 3A 27 18 81 3D 27 07 31 21 5A 27 7A :FE
45E0 20 ED C1 06 24 74 31 21 17 00 F5 C6 1F 20 37 C1 :C7
45F0 06 24 67 31 21 17 00 E8 C6 05 A6 A4 17 FE 73 27 :A6
-----------------------------------------------------------
Sum: B1 76 3D 1B 43 75 AF 66 92 4C BB E7 12 4E A1 23 :F0
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4600 0B A7 84 31 21 30 01 5A 27 50 20 EE 81 04 25 4A :8C
4610 17 00 CD 81 0D 27 35 81 3B 27 25 A6 A4 17 FE 52 :87
4620 27 38 86 20 A7 80 A6 A4 81 2F 27 31 17 FE 43 27 :FD
4630 08 A7 84 31 21 30 01 20 ED 17 00 A4 81 0D 27 0C :3F
4640 86 3B A7 80 31 21 A6 A4 81 0D 26 F6 86 0D A7 80 :E8
4650 31 21 86 00 39 35 03 16 FA D3 16 FC 9E A7 80 31 :34
4660 21 A6 A4 81 0D 27 F3 81 2F 26 F2 A7 80 31 21 20 :74
4670 C8 8D 1F 10 26 FC 77 16 FA B3 34 20 8D 14 35 20 :2A
4680 10 26 FC 6A 17 FD FE 10 26 FC 63 30 1F 9F 11 16 :58
4690 FE 91 17 FD F0 26 46 34 10 30 1F 9F 11 17 FE 5E :B5
46A0 35 10 26 39 DE 15 A6 A4 81 0D 27 05 17 FD D6 26 :AB
46B0 2C 9F 11 34 40 17 FE 46 35 40 26 21 1F 30 93 15 :5E
46C0 2A 1B 9E 15 34 10 17 FE 2C 35 10 10 9E 13 9F 13 :35
46D0 1F 20 93 13 27 07 1F 02 17 FE 9C 4F 39 86 01 39 :2D
46E0 A6 A4 81 20 26 04 31 21 20 F6 39 96 05 0F 1F A1 :00
46F0 32 27 01 39 34 30 0D 10 27 04 86 01 97 1F 17 00 :93
-----------------------------------------------------------
Sum: 81 81 48 69 6D 1A 4C 4F EA 1C 08 0D C7 C9 58 3C :14
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4700 8E 10 26 00 08 0D 28 27 02 0F 1F 35 B0 0F 1F 35 :A0
4710 30 35 06 0D 10 10 27 FA 14 30 01 17 FC C4 17 FB :E7
4720 E7 16 FA 09 00 00 00 00 00 00 00 00 00 00 00 00 :00
4730 A6 C4 84 01 27 04 17 03 24 39 35 06 16 F9 C1 00 :9C
4740 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
4750 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
4760 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
4770 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
4780 16 00 0F 16 00 C4 16 00 DB 16 07 B5 16 08 24 16 :1A
4790 00 D6 86 31 97 05 10 9E 39 9E 37 6F 84 8E 00 00 :66
47A0 9F 06 9E 06 9F 08 34 30 17 00 BD 35 30 10 26 F9 :BC
47B0 7C A6 A4 81 20 27 07 81 3B 27 03 17 00 39 8D 0D :65
47C0 A6 A0 81 0D 26 FA A6 A4 26 D8 16 F9 60 34 70 DE :2D
47D0 06 0D 25 27 0A 1F 30 D3 26 1F 03 DF 06 35 F0 30 :0D
47E0 8D F8 5C 9C 3B 27 F4 96 05 81 32 26 04 A6 84 A7 :1C
47F0 C4 30 01 33 41 20 EC 34 20 17 06 03 35 20 26 18 :7C
-----------------------------------------------------------
Sum: 79 76 84 E8 41 79 7D B4 11 E2 A4 C3 2B DA D8 19 :96
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4800 34 20 A6 A0 17 F8 FC 27 05 17 F8 FA 20 F4 86 0D :81
4810 17 F8 F3 35 20 86 01 39 34 20 9E 37 C6 08 A6 84 :38
4820 27 03 3A 20 F9 C6 06 A6 A4 81 3A 27 0B 81 3D 27 :65
4830 07 A7 80 31 21 5A 2C EF 5D 27 05 6F 80 5A 20 F8 :DF
4840 DE 08 EF 81 6F 84 35 20 4F 39 86 32 97 05 8E 00 :08
4850 00 9F 06 96 0F 27 02 97 1D 10 9E 39 8E 00 00 9F :3B
4860 11 16 F8 A5 31 8D F8 58 30 8D F7 D3 0F 24 0F 28 :C3
4870 0F 25 C6 28 6F 80 5A 26 FB 30 8D F7 C2 9F 3B A6 :82
4880 A4 81 3B 27 20 81 3D 10 27 00 95 17 F8 7E 27 04 :E9
4890 31 21 20 EB 31 21 17 05 48 27 0B 17 05 52 10 27 :EA
48A0 06 1A 16 F8 5B 39 1F 13 A6 A4 81 0D 10 27 01 09 :0D
48B0 81 3B 10 27 01 03 A6 C4 81 FF 27 78 81 FE 10 27 :36
48C0 00 81 81 FD 10 27 00 AF 81 FC 10 27 00 C8 81 FB :DD
48D0 27 4E 81 FA 10 27 00 90 17 01 8F A6 C4 84 10 10 :6C
48E0 26 00 EC 31 21 34 20 17 01 70 35 20 81 2C 10 27 :79
48F0 02 19 A6 A4 81 23 10 27 01 86 81 3C 10 27 01 C3 :7F
-----------------------------------------------------------
Sum: 22 83 1B 07 DE D9 01 93 01 A2 1A D8 4A 33 4B 6D :DC
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4900 81 5B 10 27 01 DC A6 C4 84 02 27 96 A6 43 17 01 :9E
4910 3F 17 03 9C 1F 10 17 01 1E 1F 98 17 01 19 4F 39 :CA
4920 31 21 A6 A4 81 2A 26 01 39 17 03 DB 10 26 F7 D0 :99
4930 9F 08 4F 39 31 21 17 03 CE 10 26 F7 C3 9F 06 9F :9D
4940 08 4F 39 C6 14 31 21 34 04 17 03 BB 35 04 10 26 :38
4950 F7 AE 34 04 1F 10 1F 98 35 04 17 00 F3 A6 A4 5A :AA
4960 27 04 81 2C 27 DF 4F 39 86 01 97 25 31 21 17 03 :15
4970 96 26 59 9F 26 4F 39 C6 14 31 21 A6 A0 81 2F 26 :AA
4980 4B A6 A0 81 2F 27 0D 5D 27 42 81 0D 27 3E 17 00 :45
4990 BF 5A 20 ED 4F 39 C6 0A 31 21 0F 24 5D 27 2D 34 :E8
49A0 04 17 03 0C 1F 10 17 00 8E 1F 98 17 00 89 A6 A4 :9F
49B0 35 04 5A 81 2C 27 E1 4F 39 A6 C4 84 20 27 0D A6 :B8
49C0 C4 84 10 26 07 A6 41 17 00 86 4F 39 16 F7 31 0F :DE
49D0 1B A6 41 17 00 7A 81 1F 27 22 81 1E 27 1E 31 21 :B2
49E0 17 04 29 26 E7 A6 84 81 FF 27 E1 9A 1B 97 1B A6 :10
49F0 A4 81 2C 27 E9 96 1B 17 00 56 4F 39 31 21 17 04 :74
-----------------------------------------------------------
Sum: 29 8C 12 BA F2 99 EE 18 C1 E2 A6 FB A0 4F E8 AA :D7
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4A00 0B 26 C9 A6 01 48 48 48 48 97 1B A6 A4 81 2C 26 :90
4A10 BB 31 21 17 03 F6 26 B4 A6 01 9A 1B 1F 89 17 00 :12
4A20 2F 1F 98 84 80 27 08 1F 98 84 08 27 9F 4F 39 1F :C9
4A30 98 84 08 26 97 4F 39 34 12 96 24 27 05 9E 3B A7 :15
4A40 88 14 35 12 20 0A 34 04 D6 1C 35 04 27 02 8A 10 :33
4A50 34 10 9E 3B A7 80 9F 3B 35 90 A6 A4 17 F6 A4 27 :05
4A60 08 81 2C 27 04 31 21 20 F1 39 A6 C4 84 80 27 05 :16
4A70 86 11 8D DC 39 A6 C4 84 40 27 04 86 10 8D D1 39 :BF
4A80 A6 C4 84 08 26 09 A6 C4 84 04 26 1D 86 01 39 A6 :C0
4A90 41 17 FF BC 31 21 17 02 17 10 26 00 09 1F 10 1F :22
4AA0 98 17 FF 93 4F 39 16 F6 57 A6 41 17 FF A2 31 21 :1D
4AB0 17 01 FD 10 26 FF EF 1F 10 17 FF 7B 1F 98 17 FF :C6
4AC0 76 4F 39 A6 C4 84 02 10 27 00 90 A6 42 17 FF 80 :33
4AD0 31 21 17 01 DB 10 26 00 82 1F 10 1F 98 17 FF 57 :50
4AE0 4F 39 31 21 A6 C4 84 01 27 71 A6 44 17 FF 61 86 :48
4AF0 FF 17 FF 5C 17 01 B9 26 62 A6 A4 81 5D 26 5C 1F :93
-----------------------------------------------------------
Sum: 62 63 15 42 47 D0 8E 44 08 C5 DC 3A 34 A9 29 C2 :B0
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4B00 10 17 FF 33 1F 98 17 FF 2E 4F 39 0F 1C A6 C4 84 :F5
4B10 01 27 48 A6 44 17 FF 38 A6 A4 81 2C 10 27 01 1D :F4
4B20 81 5B 26 0B 17 00 2D A6 A4 81 2C 10 27 01 0E 17 :A5
4B30 02 DA 26 2A A6 02 81 FF 27 21 97 1B 31 21 17 02 :B9
4B40 CB 26 18 A6 03 81 FF 27 12 81 8C 27 0E 9A 1B 17 :79
4B50 FE F4 4F 39 86 01 97 1C 31 21 39 16 F5 A2 34 20 :40
4B60 0F 1E 17 FE F5 31 21 17 02 A2 35 20 26 ED A6 03 :55
4B70 81 8C 27 5D 81 FF 27 E3 97 1B 17 01 8A 27 11 A6 :4D
4B80 A4 81 3C 26 04 97 1E 31 21 17 01 24 10 26 F5 70 :69
4B90 17 02 C9 96 21 81 02 27 25 81 01 27 11 96 1C 26 :FA
4BA0 0D 1F 10 C4 1F DA 1B 1F 98 17 FE A4 4F 39 96 1B :BD
4BB0 8A 88 17 FE 91 1F 10 1F 98 17 FE 94 4F 39 96 1B :80
4BC0 8A 89 17 FE 81 1F 10 17 FE 86 1F 98 17 FE 81 4F :0F
4BD0 39 0F 1E 0F 20 A6 A4 81 3C 26 04 97 20 31 21 17 :E6
4BE0 01 25 27 4A 17 02 B5 30 1E 4D 27 03 17 03 4B A6 :35
4BF0 C4 84 C0 27 02 30 1F 0F 0C 17 02 60 96 20 81 3C :87
-----------------------------------------------------------
Sum: C7 A2 80 44 AE 6B 75 86 55 CA D8 D9 DA BF 9B AE :F3
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4C00 27 13 30 1E 86 8D 17 FE 3D 1F 10 17 FE 29 1F 98 :11
4C10 17 FE 24 4F 39 96 21 30 1F 81 02 25 03 17 03 1A :A6
4C20 86 8C 17 FE 21 1F 10 1F 98 17 FE 0B 4F 39 17 02 :EF
4C30 2B 0D 20 26 EB 96 21 81 02 24 C9 20 E3 31 21 A6 :8B
4C40 A4 81 2D 27 3E 17 01 C4 26 4B A6 03 81 FF 27 45 :99
4C50 81 8C 27 41 97 1B A6 A4 81 2B 27 09 96 1B 8A 84 :0C
4C60 17 FD E3 4F 39 31 21 A6 A4 81 2B 27 0D 96 1C 26 :D3
4C70 24 96 1B 8A 80 17 FD D8 4F 39 96 1B 8A 81 17 FD :23
4C80 C5 4F 39 31 21 A6 A4 81 2D 27 0D 96 1C 26 06 86 :2F
4C90 82 97 1B 20 09 16 F4 68 31 21 86 83 97 1B 17 01 :F4
4CA0 6B 26 F2 A6 03 81 FF 27 EC 9A 1B 17 FD 98 4F 39 :A8
4CB0 8E 00 00 9F 0A 0F 0C 8D 1C 1F 10 D3 0A DD 0A 0D :FB
4CC0 24 27 04 86 01 97 28 A6 A4 81 2B 27 EA 81 2D 27 :71
4CD0 E6 9E 0A 4F 39 8E 00 00 0F 23 A6 A4 81 2B 26 02 :F4
4CE0 31 21 81 2D 26 06 86 01 97 23 31 21 8D 4F 27 16 :D8
4CF0 96 05 81 32 10 27 00 BC 8E 00 00 A6 A4 17 F4 0C :30
-----------------------------------------------------------
Sum: 60 41 33 9C 00 F0 7F B4 CE D3 27 4A 37 A3 22 5E :FF
```

アセンブラ ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4D00 27 04 31 21 20 F5 39 8E 00 00 9F 0A 0F 23 0F 0C :4F
4D10 A6 A4 81 2B 26 02 31 21 81 2D 26 06 31 21 86 01 :23
4D20 97 23 8D 19 26 14 1F 10 D3 0A DD 0A A6 A4 81 2B :83
4D30 27 DE 81 2D 27 DA 9E 0A 4F 39 86 01 39 34 20 5F :57
4D40 8E 00 00 A6 A4 81 27 27 5A 81 24 10 27 F3 C0 80 :10
4D50 30 25 4B 81 0A 24 47 20 0F A6 A4 17 F3 AE 27 21 :0F
4D60 80 30 25 3A 81 0A 24 36 34 06 1F 10 58 49 DD 19 :F4
4D70 58 49 58 49 D3 19 1F 01 35 06 1F 89 3A 31 21 20 :DD
4D80 D8 0D 23 27 11 1F 10 43 53 C3 00 01 1F 01 84 80 :ED
4D90 27 0C 35 06 4F 39 1F 10 84 80 26 02 20 F4 35 20 :BA
4DA0 86 01 39 1F 10 4F 31 21 E6 A0 1F 01 35 06 86 01 :F8
4DB0 97 0C 4F 39 86 01 97 0C A6 A4 81 2A 27 08 8D 3F :45
4DC0 27 09 86 01 97 24 9E 06 16 FF 30 A6 84 E6 01 1F :8B
4DD0 01 0D 23 10 27 FF 24 1F 10 43 53 C3 00 01 16 FF :29
4DE0 1A CC 00 09 DD 29 30 8D 02 16 C6 04 D7 22 20 2B :D8
4DF0 CC 00 06 DD 29 30 8D 05 D7 C6 04 D7 22 20 1C CC :3C
---------------------------------------------------------
Sum: 4B 4F 17 B8 4F D1 4E 7E D7 48 41 4D E3 63 3A 66 :E8


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4E00 00 08 DD 29 9E 37 C6 06 D7 22 20 0F CC 00 06 DD :86
4E10 29 30 8D 06 9B C6 02 D7 22 20 00 10 9F 35 A6 84 :76
4E20 27 34 34 30 D6 22 A6 84 27 17 A1 A4 26 09 31 21 :E5
4E30 30 01 5A 27 0C 20 EF 35 30 1F 10 D3 29 1F 01 20 :9D
4E40 DA A6 A4 17 F2 C6 26 EF 5D 27 05 5A 30 01 20 F8 :34
4E50 35 06 35 06 4F 39 10 9E 35 86 01 39 96 0C 26 31 :9A
4E60 1F 10 4D 27 09 81 FF 27 19 86 02 97 21 39 1F 10 :14
4E70 C4 80 26 F5 1F 10 C4 F0 26 03 0F 21 39 86 01 97 :F2
4E80 21 39 1F 10 C4 80 27 E1 1F 10 53 C4 F0 26 EE 20 :3F
4E90 E9 1F 10 4D 26 D3 0D 1E 27 CF 20 E1 8E 00 00 0F :1D
4EA0 24 96 05 81 32 26 0B 17 FE 06 26 08 1F 10 93 06 :B4
4EB0 1F 01 4F 39 8E 00 00 86 01 97 24 39 1F 13 17 FB :F5
4EC0 A9 A6 41 17 FB 8A 31 21 17 FE 3C 26 14 A6 C4 84 :F7
4ED0 02 27 05 1F 10 17 FB 78 1F 10 1F 98 17 FB 71 4F :9F
4EE0 39 96 05 81 32 26 E6 17 FF B2 A6 C4 84 40 27 02 :B2
4EF0 30 1F 30 1F 30 1F A6 C4 84 02 26 30 1F 10 81 FF :E2
---------------------------------------------------------
Sum: D3 1A 42 AC 9B 2E 4D 4A 1F EC CC 79 64 63 B9 76 :81


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4F00 27 18 4D 27 03 17 00 32 1F 10 C4 80 27 03 17 00 :B3
4F10 29 1F 10 1F 98 17 FB 1F 4F 39 1F 10 C4 80 26 03 :64
4F20 17 00 17 1F 10 1F 98 17 FB 0D 4F 39 30 1F 1F 10 :39
4F30 17 FB 04 1F 98 17 FA FF 4F 39 86 01 97 24 97 28 :66
4F40 39 96 05 81 32 27 01 39 34 30 86 12 97 22 86 20 :43
4F50 17 F1 B3 9E 06 17 F1 BA 86 20 17 F1 A9 86 0C 97 :A1
4F60 22 30 8D F0 DA C6 05 9C 3B 27 20 5D 27 1D A6 88 :61
4F70 14 26 07 A6 80 17 00 29 20 0A 86 2D 17 F1 87 17 :2A
4F80 F1 84 30 01 0A 22 0A 22 5A 20 DC 9F 3D 0D 22 27 :86
4F90 09 86 20 17 F1 70 0A 22 20 F1 17 F8 30 35 30 4F :57
4FA0 39 34 04 17 F1 6F 96 0D 17 F1 5B 96 0E 17 F1 56 :F0
4FB0 35 04 39 96 05 34 10 81 32 27 02 35 90 9E 3D 9C :69
4FC0 3B 26 02 35 90 C6 0B 17 F1 4E C6 05 9E 3D 9C 3B :CC
4FD0 27 25 5D 27 19 A6 88 14 26 07 A6 80 17 FF C2 20 :76
4FE0 0A 86 2D 17 F1 20 17 F1 1D 30 01 5A 20 E0 9F 3D :71
4FF0 86 0D 17 F1 11 20 CE 9F 3D 86 0D 17 F1 08 35 90 :DE
---------------------------------------------------------
Sum: 5E 2F F4 62 71 60 B6 AC 01 44 C5 AF 01 97 64 21 :EC


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5000 41 42 58 00 20 3A 00 00 00 41 44 43 41 0B 89 99 :6B
5010 B9 A9 41 44 43 42 0B C9 D9 F9 E9 41 44 44 41 0B :10
5020 8B 9B BB AB 41 44 44 42 0B CB DB FB EB 41 44 44 :F7
5030 44 07 C3 D3 F3 E3 41 4E 44 41 0B 84 94 B4 A4 41 :87
5040 4E 44 42 0B C4 D4 F4 E4 41 4E 44 43 08 1C 00 00 :89
5050 00 41 53 4C 41 20 48 00 00 00 41 53 4C 42 20 58 :23
5060 00 00 00 41 53 4C 00 03 00 08 78 68 41 53 52 41 :F2
5070 20 47 00 00 00 41 53 52 42 20 57 00 00 00 41 53 :9A
5080 52 00 03 00 07 77 67 42 49 54 41 0B 85 95 B5 A5 :D9
5090 42 49 54 42 0B C5 D5 F5 E5 43 4C 52 41 20 4F 00 :31
50A0 00 00 43 4C 52 42 20 5F 00 00 00 43 4C 52 00 03 :86
50B0 00 0F 7F 6F 43 4D 50 41 0B 81 91 B1 A1 43 4D 50 :6D
50C0 42 0B C1 D1 F1 E1 43 4D 50 44 47 83 93 B3 A3 43 :CB
50D0 4D 50 58 07 8C 9C BC AC 43 4D 50 59 47 8C 9C BC :F0
50E0 AC 43 4D 50 53 87 8C 9C BC AC 43 4D 50 55 87 83 :35
50F0 93 B3 A3 43 4F 4D 41 20 43 00 00 00 43 4F 4D 42 :8D
---------------------------------------------------------
Sum: 99 02 CE C2 B5 40 97 1E 76 11 5F 7B B9 22 C9 D1 :AB


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5100 20 53 00 00 00 43 4F 4D 00 03 00 03 73 63 43 57 :C8
5110 41 49 20 3C 00 00 00 44 41 41 00 20 19 00 00 00 :E5
5120 44 45 43 41 20 4A 00 00 00 44 45 43 42 20 5A 00 :FF
5130 00 00 44 45 43 00 03 00 0A 7A 6A 45 4F 52 41 0B :EF
5140 88 98 B8 A8 45 4F 52 42 0B C8 D8 F8 E8 45 58 47 :17
5150 00 10 1E 00 00 00 49 4E 43 41 20 4C 00 00 00 49 :FE
5160 4E 43 42 20 5C 00 00 00 49 4E 43 00 03 00 0C 7C :B4
5170 6C 4A 4D 50 00 03 00 0E 7E 6E 4A 53 52 00 03 00 :42
5180 9D BD AD 4C 44 41 00 0B 86 96 B6 A6 4C 44 42 00 :2D
5190 0B C6 D6 F6 E6 4C 44 44 00 07 CC DC FC EC 4C 44 :7E
51A0 58 00 07 8E 9E BE AE 4C 44 59 00 47 8E 9E BE AE :BF
51B0 4C 44 53 00 47 CE DE FE EE 4C 44 55 00 07 CE DE :5A
51C0 FE EE 4C 45 41 58 01 00 00 00 30 4C 45 41 59 01 :73
51D0 00 00 00 31 4C 45 41 53 01 00 00 00 32 4C 45 41 :5B
51E0 55 01 00 00 00 33 4C 53 4C 41 20 48 00 00 00 4C :69
51F0 53 4C 42 20 58 00 00 00 4C 53 4C 00 03 00 08 78 :C7
---------------------------------------------------------
Sum: D9 18 77 40 F8 C8 4B 6E B1 9D 96 F4 AA 7C 05 44 :68


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5200 68 4C 53 52 41 20 44 00 00 00 4C 53 52 42 20 54 :A5
5210 00 00 00 4C 53 52 00 03 00 04 74 64 4D 55 4C 00 :BE
5220 20 3D 00 00 00 4E 45 47 41 20 40 00 00 00 4E 45 :6B
5230 47 42 20 50 00 00 00 4E 45 47 00 03 00 00 70 60 :A6
5240 4E 4F 50 00 20 12 00 00 00 4F 52 41 00 0B 8A 9A :30
5250 BA AA 4F 52 42 00 0B CA DA FA EA 4F 52 43 43 08 :09
5260 1A 00 00 00 50 53 48 53 10 34 00 00 00 50 53 48 :87
5270 55 10 36 00 00 00 50 55 4C 53 10 35 00 00 00 50 :74
5280 55 4C 55 10 37 00 00 00 52 4F 4C 41 20 49 00 00 :D4
5290 00 52 4F 4C 42 20 59 00 00 00 52 4F 4C 00 03 00 :98
52A0 09 79 69 52 4F 52 41 20 46 00 00 00 52 4F 52 42 :BA
52B0 20 56 00 00 00 52 4F 52 00 03 00 06 76 66 52 54 :F4
52C0 49 00 20 3B 00 00 00 52 54 53 00 20 39 00 00 00 :F6
52D0 53 42 43 41 0B 82 92 B2 A2 53 42 43 42 0B C2 D2 :45
52E0 F2 E2 53 45 58 00 20 1D 00 00 00 53 54 41 00 03 :EC
52F0 00 97 B7 A7 53 54 42 00 03 00 D7 F7 E7 53 54 44 :81
---------------------------------------------------------
Sum: 52 FC C2 56 C4 BF 09 9D 4D 33 03 C2 DB D2 07 E2 :6A


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5300 00 03 00 DD FD ED 53 54 58 00 03 00 9F BF AF 53 :2C
5310 54 59 00 43 00 9F BF AF 53 54 53 00 43 00 DF FF :18
5320 EF 53 54 55 00 03 00 DF FF EF 53 55 42 41 0B 80 :71
5330 90 B0 A0 53 55 42 42 0B C0 D0 F0 E0 53 55 42 44 :A5
5340 07 83 93 B3 A3 53 57 49 00 20 3F 00 00 00 53 57 :6F
5350 49 32 60 3F 00 00 00 53 57 49 33 A0 3F 00 00 00 :1F
5360 53 59 4E 43 20 13 00 00 00 54 46 52 00 10 1F 00 :8B
5370 00 00 54 53 54 41 20 4D 00 00 00 54 53 54 42 20 :06
5380 5D 00 00 00 54 53 54 00 03 00 0D 7D 6D 4F 52 47 :3A
5390 00 FF 00 00 00 00 46 43 42 00 FE 00 00 00 00 46 :0E
53A0 43 43 00 FD 00 00 00 00 46 44 42 00 FC 00 00 00 :4B
53B0 00 45 51 55 00 FB 00 00 00 00 52 4D 42 00 FA 00 :C1
53C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
53D0 42 43 43 00 00 24 42 43 53 00 00 25 42 45 51 00 :C1
53E0 00 27 42 47 45 00 00 2C 42 47 54 00 00 2E 42 48 :B6
53F0 49 00 00 22 42 4C 45 00 00 2F 42 4C 53 00 00 23 :71
---------------------------------------------------------
Sum: A1 5E 5F 0B 44 36 EC 88 E1 8A 86 B6 49 7B 6E 85 :B5


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5400 42 4C 54 00 00 2D 42 4D 49 00 00 2B 42 4E 45 00 :E7
5410 00 26 42 50 4C 00 00 2A 42 52 41 00 00 20 42 52 :B7
5420 4E 00 00 21 42 53 52 00 00 8D 42 56 43 00 00 28 :E6
5430 42 56 53 00 00 29 4C 42 43 43 42 24 4C 42 43 53 :B2
5440 42 25 4C 42 45 51 42 27 4C 42 47 45 42 2C 4C 42 :0A
5450 47 54 42 2E 4C 42 48 49 42 22 4C 42 4C 45 42 2F :1E
5460 4C 42 4C 53 42 23 4C 42 4C 54 42 2D 4C 42 4D 49 :53
5470 42 2B 4C 42 4E 45 42 26 4C 42 50 4C 42 2A 4C 42 :1A
5480 52 41 02 16 4C 42 52 4F 42 21 4C 42 53 52 02 17 :89
5490 4C 42 56 43 42 28 4C 42 56 53 42 29 00 00 00 00 :33
54A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
54B0 41 00 02 08 86 FF 42 00 04 09 85 FF 44 00 FF 00 :E6
54C0 8B FF 58 00 10 01 FF 00 59 00 20 02 FF 20 53 00 :DF
54D0 40 04 FF 60 55 00 40 03 FF 40 43 43 01 0A FF FF :09
54E0 50 43 80 05 FF 8C 44 50 08 0B FF FF 00 00 00 00 :48
54F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
---------------------------------------------------------
Sum: E3 77 40 3C 27 9A 5B 75 F0 E4 5F 53 84 09 44 DF :9D
```

FM-8

エディタ・アセンブラのソースをジェネレート！

2 PASS DISASSEMBLR

■馬場二郎

逆アセンブラは6809においてもかなり出回っていますが，案外２パス形式のものは少ないようです．アセンブラと違って解析には１パスでも充分ということなのでしょうか．私もこれまでBASICで書かれた逆アセンブラを使っていますが，スピードが遅い，JMP先をさがすのが面倒，オフセットがわかりにくいなどかなり不満があります．

特徴

❶ すべてマシン語で高速，省スペース．

❷ ２パス方式でラベルが自動作成され，さらに３種のパターンが指定できる．

❸ ラベルの追加，削除を行なう簡単なエディタを持つ．

❹ ２パス・アセンブラの完全なソースとして出力でき，しかも使っている命令テーブルが同一であるため，ＯＰコードの不一致はまず起こりません．

❺ 256単位（＄××００）でポジション・インディペンデントです．

❻ ダイレクト・ページの値を設定でき，値を正しく指定すれば正確なラベルを与えます．

❼ オフセットの表示が何ビット・モードか明確にしています．

このプログラムは，ＯＰコード・テーブルと入出力などいくつかのサブルーチンを２パス・アセンブラから借用しているので，2.5K弱と2パスとしては小さくまとまりました．アセンブラと合せても８K弱ですから充分両者を常駐して使用できます．

❷のパターンというのはラベルの作成に次の制限を与え，場合により不必要なラベルの作成をやめようというものです．

オプション

0または省略	逆アセンブル対象内のアドレスのラベルのみ
1	上およびBRAやPC相対のような相対指定のもの
2	ダイレクト，エクステンドを含みすべて

当初０を省略モードにしましたが，これは２パス・アセンブラのソースとして出力したとき未登録ラベルとならないためのものでした．後で範囲外のラベルもソースに落とすようにしましたので，オプション１がもっとも見やすいと思います．

オフゼットの表示については，５ビットを符号付10進とし，８ビットを'<'＋2桁の16進，16ビットを４桁の16進としました．これはアセンブラの仕様に合わせたものですが，実際５ビットは10進指定が一般的ですし，見やすいと思います．逆に16ビットは10進で使われることはまずないと思います．次に例を示しますが，これらの表示によってアセンブルのソースとして使った場合も同一のオブジェクトを与えることができるわけです．

A6	84			→	LDA	,X
A6	00			→	LDA	0,X
A6	88	00		→	LDA	<$00,X
A6	89	00	00	→	LDA	$0000,X

未定義の命令が見つかると，'FCB'と16進で出力されます．ASCIIダンプはありませんが，ソースとして出力するのでどこまで出力してよいのかわからないのであえて付けませんでした．

注意

ソースの上限とラベルの上限のチェックを一切していないので，あまり無茶な領域の設定はしないでください．目安としては現在のポジションで２～３Kぐらいです．長いものを解読したい人は，＄6037-38を空いているRAMしてください．ソースとラベルの領域は，アセンブラとまったく同一にしています．それと，BASICでCLEARしておいたほうがよいですよ．

使い方

BASICからEXEC & H5600 CR または，MONモードでG5600 CR でシステムの状態リポートとともにコマンド待ちになります（'>'－表示）．

ラベルはAコマンドで，追加を行ないます．いったんクリアするのではありません．複数の領域を逆アセンブルする場合に，ラベルが領域（上限チェックはしていません）をオーバーしないようにしてください．Aコマンドにオプションが設けてありますが，長いプログラムではラベル領域が不足する，不必要なもの（定数として使っているもの）まで行なってしまう，２パス・アセンブルのとき未登録ラベルとなってしまう，ということで，場合により行なえるようにしました．

Eコマンドはラベルの追加，削除を1単位で行う簡単なエディタです．エディタ・モードは('ー')のみとなります．

追　加	××××ーLABEL CR
削　除	×××× CR
コマンドへの戻り	CR のみ

Tコマンドは登録されているラベルを出力します．途中リストを止めたいときには，スペース・バーを押し，続けるときは再度スペース・バーを，コマンドへ移るときは他のキー（何でも良い）を押します．

Lコマンドは逆アセンブルを指定したアドレスから始めます．アドレスを省略すれば，Wコマンドで指定したアドレスより始めます．リストを止めたいときはTコマンドのときと同じです．

ラベルをまったく登録しなくてもLコマンドは実行できますが，多くの1パス・アセンブラのように相対アドレスを絶対アドレスに変換してはくれません(やるのは簡単ですが，あくまでアセンブル可能なソース出力を前提としています)．

ダイレクト・ページは通常0に設定されていますが，本ディスアセンブラでは指定してやることができ，従ってダイレクト・モードも適切なアドレスを設定してやれば正常なラベルを与えます．

ラベルの形式は'<'＋16進4桁で与え，ダイレクト・モードまたはPC相対8ビットは前に'<'を加えてリストします．ラベルは6文字までです．

プログラム

自分としては満足しておりますが，私もそうですが，どうも人の作ったものは合わないと思う人が多いと思うので，簡単にロジックの説明をします．コンパクトにまとまっていると思いますが，JMPテーブルは一切使っていないので，リストを追っていけばだいたい解析できると思います．

命令テーブルは2パス・アセンブラのものを使っています．構成は次のようになっています．

```
AB×00   ×     3A  00  00  00
└──┘
オペコード    └→ビット　7：2バイト・オペ($11) 5：インヘ
                 レント 3：イミーディエイト 1：ダイレ
                 クト/エクステンド 6：2バイト・オペ
                 ($10) 4：インヘレント(2バイト)
                 2：イミーディエイト(2バイト) 0：イ
                 ンデックス
BRA  00×20
          └→ビット7：2バイト・オペ($11) 1：ロング・ブランチ
                 6：1バイト・オペ($10)
```

これによって，命令サーチの段階で，各モードへ分岐します．

リスト出力は分析中に行なうのではなく，いったんワーク・バッファに格納し，TABなどはリスト専用のルーチンで処理します．オブジェクトもオブジェクト・バッファに入ります．したがってもし出力フォームが気にいらないときは，リスト・レーチンを，ASCIIダンプなどがコメントとして必要なときは，各処理ルーチン中に加えてください．このとき，(A) 1文字をテキスト・バッファに追加するルーチン（$08で1行分削除も行える）と，(X)から（B）の数または（X）＝0となったデータまでテキストを追加するルーチンがあります．

〈コマンド表〉

コマンド	機　能
B	MONモードへ
C	ラベル領域クリア
E	ラベル追加，削除
L［××××］	逆アセングル・リスト
P	プリンタ・スイッチ(ON/OFF)
R	システム・リポート
S	ソース出力スイッチ(ON/OFF)
T	ラベル登録リスト
W××××，××××	アセンブル対象範囲の設定
A［0］	ラベルの自動作成 0：対象内のラベル 1：同上および相対指定のすべて 2：ダイレクトを含めすべて
D××	ダイレクト・ページ・レジスタの設定

もっとも仕様の決定と発表のための整備のほうがコーディングより長かったくらいで（デバックは実質半日で非常に調子良くいきました）細かい点は自分で解析してください．そのための道具ですよ．

反省

いまだに80時代の考え方しかできず，どうしてもXをHLレジスタYをDEレジスタBまたはDをBCレジスタといった具合にしか考えられず，UやSを効率的に使っているとは言えません．本当は2Kでおさえるつもりでしたが….

参考文献

1）〝Z80逆アセンブラ〞インターフェース，1980．4

2）6809ハンドブック，アスキー出版

①ラベル未登録で出力

```
>W6248, 6282 RETURN
>L RETURN

0001  6248 810D          CMPA #$0D
0002  624A 2609          BNE  $09
0003  624C 8D07          BSR  $07
0004  624E 860A          LDA  #$0A
0005  6250 8D03          BSR  $03
0006  6252 860D          LDA  #$0D
0007  6254 39            RTS
0008  6255
0009  6255 3402          PSHS A
000A  6257 961D          LDA  <$1D
000B  6259 3502          PULS A
000C  625B 2702          BEQ  $02
000D  625D 8D64          BSR  $64
000E  625F 0D1F          TST  <$1F
000F  6261 2602          BNE  $02
0010  6263 8D01          BSR  $01
0011  6265 39            RTS
0012  6266
0013  6266 3456          PSHS A,B,X,U
0014  6268 308DFDBF      LEAX $FDBF,PC
0015  626C 338DFDC4      LEAU $FDC4,PC
0016  6270 A7C4          STA  ,U
0017  6272 CC1400        LDD  #$1400
0018  6275 ED84          STD  ,X
0019  6277 EF02          STU  2,X
001A  6279 CC0001        LDD  #$0001
001B  627C ED04          STD  4,X
001C  627E ADFFFBFA      JSR  [$FBFA]
001D  6282 35D6          PULS A,B,X,U,PC
001E  6284
```

ブランチなどが絶対番地になっていません．これでは見にくいですね．

RTSとPULSでPCを含むとき次の行に出力します．

②オプション0でラベルを登録し出力

```
0001  6248 810D           CMPA #$0D
0002  624A 2609           BNE  L6255
0003  624C 8D07           BSR  L6255
0004  624E 860A           LDA  #$0A
0005  6250 8D03           BSR  L6255
0006  6252 860D           LDA  #$0D
0007  6254 39             RTS
0008  6255               ;
0009  6255 3402    L6255 :PSHS A
000A  6257 961D           LDA  <$1D
000B  6259 3502           PULS A
000C  625B 2702           BEQ  L625F
000D  625D 8D64           BSR  $64  <-----
000E  625F 0D1F    L625F :TST  <$1F
000F  6261 2602           BNE  L6265
0010  6263 8D01           BSR  L6266
0011  6265 39      L6265 :RTS
0012  6266               ;
0013  6266 3456    L6266 :PSHS A,B,X,U
0014  6268 308DFDBF       LEAX $FDBF,PC
0015  626C 338DFDC4       LEAU $FDC4,PC
0016  6270 A7C4           STA  ,U
0017  6272 CC1400         LDD  #$1400
0018  6275 ED84           STD  ,X
0019  6277 EF02           STU  2,X
001A  6279 CC0001         LDD  #$0001
001B  627C ED04           STD  4,X
001C  627E ADFFFBFA       JSR  [$FBFA]
001D  6282 35D6           PULS A,B,X,U,PC
001E  6284
```

対象範囲外なのでラベル付されない。

③オプション1でラベルを登録し出力

```
0001  6248                L62C3 =$62C3
0002  6248                L602B =$602B
0003  6248                L6034 =$6034
0004  6248 810D                 CMPA #$0D
0005  624A 2609                 BNE  L6255
0006  624C 8D07                 BSR  L6255
0007  624E 860A                 LDA  #$0A
0008  6250 8D03                 BSR  L6255
0009  6252 860D                 LDA  #$0D
000A  6254 39                   RTS
000B  6255                      ;
000C  6255 3402          L6255 :PSHS A
000D  6257 961D                 LDA  <$1D
000E  6259 3502                 PULS A
000F  625B 2702                 BEQ  L625F
0010  625D 8D64                 BSR  L62C3
0011  625F 0D1F          L625F :TST  <$1F
0012  6261 2602                 BNE  L6265
0013  6263 8D01                 BSR  L6266
0014  6265 39            L6265 :RTS
0015  6266                      ;
0016  6266 3456          L6266 :PSHS A,B,X,U
0017  6268 308DFDBF             LEAX L602B,PC
0018  626C 338DFDC4             LEAU L6034,PC
0019  6270 A7C4                 STA  ,U
001A  6272 CC1400               LDD  #$1400
001B  6275 ED84                 STD  ,X
001C  6277 EF02                 STU  2,X
001D  6279 CC0001               LDD  #$0001
001E  627C ED04                 STD  4,X
001F  627E ADFFFBFA             JSR  [$FBFA]
0020  6282 35D6                 PULS A,B,X,U,PC
0021  6284
```

対象範囲外のラベルを出力します。

後で付け加えた機能ですが（おかげで2.25Kでおさまらなくなった），自分でいうのもなんですが非常に便利です。

ここでは，この範囲以外に参照されているものが3つあることがすぐわかりますね。

④オプション2でラベルを登録し出力

```
0001  6248                L62C3 =$62C3
0002  6248                L602B =$602B
0003  6248                L6034 =$6034
0004  6248                L001D =$001D
0005  6248                L001F =$001F
0006  6248                L1400 =$1400
0007  6248                L0001 =$0001   <-----
0008  6248 810D                 CMPA #$0D
0009  624A 2609                 BNE  L6255
000A  624C 8D07                 BSR  L6255
000B  624E 860A                 LDA  #$0A
000C  6250 8D03                 BSR  L6255
000D  6252 860D                 LDA  #$0D
000E  6254 39                   RTS
000F  6255                      ;
0010  6255 3402          L6255 :PSHS A
0011  6257 961D                 LDA  <L001D
0012  6259 3502                 PULS A
0013  625B 2702                 BEQ  L625F
0014  625D 8D64                 BSR  L62C3
0015  625F 0D1F          L625F :TST  <L001F
0016  6261 2602                 BNE  L6265
0017  6263 8D01                 BSR  L6266
0018  6265 39            L6265 :RTS
0019  6266                      ;
001A  6266 3456          L6266 :PSHS A,B,X,U
001B  6268 308DFDBF             LEAX L602B,PC
001C  626C 338DFDC4             LEAU L6034,PC
001D  6270 A7C4                 STA  ,U
001E  6272 CC1400               LDD  #L1400
001F  6275 ED84                 STD  ,X
0020  6277 EF02                 STU  2,X
0021  6279 CC0001               LDD  #L0001
0022  627C ED04                 STD  4,X
0023  627E ADFFFBFA             JSR  [LFBFA]
0024  6282 35D6                 PULS A,B,X,U,PC
0025  6284                      ;
```

何でもラベル付けすれば良いというものではありませんね。これらは本来定数です。

ダイレクト・ページの省略時が00。

⑤エディタでラベルを意味付けDPRを$60に設定

```
L6255 :6255    L625F :625F    L6265 :6265    L6266 :6266
SW1   :601D    SW3   :601F    PRTOUT:62C3    RCB   :602B
DSPDTA:6034    BIOS  :FBFA    DSPLY :6248
```

```
0001  6248                SW1   =$601D
0002  6248                SW3   =$601F
0003  6248                PRTOUT=$62C3
0004  6248                RCB   =$602B
0005  6248                DSPDTA=$6034
0006  6248 810D          DSPLY :CMPA #$0D
0007  624A 2609                 BNE  L6255
0008  624C 8D07                 BSR  L6255
0009  624E 860A                 LDA  #$0A
000A  6250 8D03                 BSR  L6255
000B  6252 860D                 LDA  #$0D
000C  6254 39                   RTS
000D  6255                      ;
000E  6255 3402          L6255 :PSHS A
000F  6257 961D                 LDA  <SW1
0010  6259 3502                 PULS A
0011  625B 2702                 BEQ  L625F
0012  625D 8D64                 BSR  PRTOUT
0013  625F 0D1F          L625F :TST  <SW3
0014  6261 2602                 BNE  L6265
0015  6263 8D01                 BSR  L6266
0016  6265 39            L6265 :RTS
0017  6266                      ;
0018  6266 3456          L6266 :PSHS A,B,X,U
0019  6268 308DFDBF             LEAX RCB,PC
001A  626C 338DFDC4             LEAU DSPDTA,PC
001B  6270 A7C4                 STA  ,U
001C  6272 CC1400               LDD  #$1400
001D  6275 ED84                 STD  ,X
001E  6277 EF02                 STU  2,X
001F  6279 CC0001               LDD  #$0001
0020  627C ED04                 STD  4,X
0021  627E ADFFFBFA             JSR  [BIOS]
0022  6282 35D6                 PULS A,B,X,U,PC
b023  6284                      ;
```

解析してわかったらラベルに意味付けしてソースに出力しセーブしましょう。

ディスアセンブラ・ダンプ・リスト

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
5600	1C	00	30	8D	09	FF	1F	10	1F	8B	16	00	A4	0F	1D	0F	:AF
5610	1E	0F	10	86	0D	17	0C	30	86	3E	17	0C	2B	17	0C	00	:58
5620	A6	A4	81	45	10	27	01	46	81	42	27	3F	81	43	27	3E	:E0
5630	81	4C	10	27	02	AF	81	54	10	27	01	71	81	50	27	34	:5F
5640	81	52	27	6D	81	41	10	27	02	89	81	53	27	32	81	57	:F0
5650	27	3A	81	44	27	52	20	B5	8D	02	20	FA	17	0C	82	49	:0B
5660	4E	50	55	54	20	45	52	52	0D	00	39	7E	AF	A7	6F	9D	:76
5670	09	C5	20	3D	0D	0F	27	04	0F	0F	20	35	03	0F	20	31	:48
5680	0D	20	27	04	0F	20	20	29	03	20	20	25	17	0D	F6	26	:78
5690	C7	A6	A4	81	2C	26	C1	34	10	17	0D	E9	1F	13	35	10	:6D
56A0	26	B6	9F	15	DF	13	20	09	17	0D	DA	26	AB	1F	10	D7	:80
56B0	23	86	0D	17	0B	92	17	0C	28	46	4D	2D	38	20	36	38	:3B
56C0	30	39	20	44	49	53	41	53	53	45	4D	42	4C	45	52	20	:27
56D0	56	31	2E	31	0D	53	54	41	52	54	3A	00	9E	15	17	0D	:92
56E0	01	17	0B	FD	20	45	4E	44	3A	00	9E	13	17	0C	F3	17	:2F
56F0	0B	EF	20	4C	41	42	45	4C	3A	00	17	00	66	17	0C	E2	:36
Sum:	0F	12	DE	30	D9	EB	96	A2	4C	EF	DF	72	41	B9	E2	5A	:BD

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
5700	17	0B	DE	20	44	50	3A	00	96	23	17	0D	2F	96	0D	17	:B4
5710	0B	36	96	0E	17	0B	31	17	0B	C7	20	50	52	49	4E	54	:CE
5720	45	52	3A	00	0D	0F	27	08	17	0B	B6	4F	4E	00	20	07	:B8
5730	17	0B	AE	4F	46	46	00	17	0B	A7	20	53	4F	52	43	45	:10
5740	3A	00	9E	39	A6	80	26	FC	17	0C	97	0D	20	27	09	17	:87
5750	0B	8F	20	4F	4E	00	20	08	17	0B	86	20	4F	46	46	00	:22
5760	16	FE	AA	9E	37	6D	84	27	04	30	08	20	F8	39	17	0A	:59
5770	AF	86	01	97	1E	A6	A4	81	0D	27	E5	31	3F	17	0D	05	:68
5780	26	26	A6	A4	81	0D	27	7B	81	20	26	1C	17	0F	51	81	:A1
5790	0D	27	70	17	00	97	27	D6	17	0B	46	4C	41	42	45	4C	:17
57A0	20	45	52	52	0D	00	20	C6	17	FE	B1	20	C1	9E	37	0D	:85
57B0	0F	27	02	03	1D	17	01	05	86	0D	17	0A	8B	C6	04	6D	:EB
57C0	84	27	1D	8D	23	86	3A	17	0A	7E	34	10	AE	06	17	0C	:F2
57D0	11	17	07	4E	17	07	4B	35	10	30	08	5A	26	E1	20	D5	:B9
57E0	86	0D	17	0A	63	16	FE	25	34	14	C6	06	A6	80	27	06	:B7
57F0	17	0A	55	5A	26	F6	5D	27	08	86	20	17	0A	4A	5A	20	:03
Sum:	1C	BF	BF	89	65	97	4F	96	8D	88	6D	96	EC	54	BA	2B	:41

ディスアセンブラ・ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5800 F5 35 94 8D 65 26 91 1F 31 8D DD 17 0A D3 20 44 :79
5810 45 4C 45 54 45 0D 00 17 FF 49 1F 10 30 48 9F 19 :3A
5820 93 19 C3 00 01 1F 02 17 0D 4D 16 FF 41 34 60 0D :F9
5830 1E 26 08 9C 15 25 08 9C 13 22 04 8D 2D 26 04 86 :69
5840 01 35 E0 8D 38 27 F8 C6 06 A6 A4 17 0C 7D 27 07 :DE
5850 A6 A0 A7 C0 5A 26 F2 C1 05 2A E4 5D 27 05 6F C0 :AB
5860 5A 20 F8 AF C1 6F C4 4F 35 E0 DE 37 6D C4 27 0A :F0
5870 AC 46 27 04 33 48 20 F4 4F 39 86 01 39 CC 00 08 :C8
5880 DD 29 34 30 9E 37 C6 06 D7 22 17 15 8E 35 B0 0D :B0
5890 05 27 07 8D D5 26 02 1F 31 39 34 30 31 8D 07 C8 :37
58A0 86 4C A7 A4 1F 10 17 0B 93 DE 0D EF 21 1F 98 17 :CA
58B0 0B 8A DE 0D EF 23 6F 25 17 FF 72 35 B0 17 09 C4 :77
58C0 81 20 26 0E 17 09 BD 27 FB 81 20 27 05 35 06 16 :F2
58D0 FD 3B 39 0F 05 A6 21 81 31 26 02 97 10 81 32 26 :A6
58E0 6B 97 10 20 67 86 32 97 05 9E 39 A6 80 26 FC 30 :3C
58F0 1F 9F 09 0D 0F 27 04 86 01 97 1D A6 21 81 0D 27 :C5
---------------------------------------------------------
Sum: 13 B2 82 35 59 67 CB CD C3 42 44 D2 C7 DC 79 0C :17
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5900 4B 17 0B 81 10 26 FD 50 20 44 0D 05 10 27 01 1B :3A
5910 0D 0B 10 26 01 15 6D C4 27 2B AE 46 9C 15 25 04 :B5
5920 9C 13 2F 1D 1F 31 C6 06 17 01 F3 86 3D 17 01 FC :F9
5930 86 01 97 21 AE 46 17 04 D0 86 0D 17 01 EE 33 48 :32
5940 39 33 48 20 D1 86 01 97 0B 16 00 DF 9E 15 9F 06 :1B
5950 8E 00 01 9F 11 DE 37 0F 0B 17 FF 61 0F 26 0F 24 :4D
5960 0F 25 30 8D 06 D9 9F 3B 0F C0 96 10 81 32 26 06 :FE
5970 86 01 97 1E 20 02 0F 1E 9E 06 9C 13 10 2E FC 8D :A5
5980 8D 88 9E 11 8D 27 30 01 9F 11 9E 06 D6 26 3A 9F :D2
5990 06 0D 24 27 C4 30 8D 06 A6 9F 3B 86 3B 97 C0 86 :03
59A0 0D 97 C1 9E 11 8D 06 30 01 9F 11 20 AC 34 50 0D :E5
59B0 05 27 76 0D 20 27 12 DE 09 31 8D 07 03 A6 A0 A7 :A4
59C0 C0 81 0D 26 F8 6F C4 DF 09 17 0A 16 17 05 53 9E :CB
59D0 06 17 0A 0E 17 05 50 30 8D 06 64 C6 05 5D 27 12 :29
59E0 9C 3B 27 08 A6 80 17 15 B8 5A 20 F1 17 05 33 5A :24
59F0 26 FA 17 05 32 31 8D 06 C7 A6 A4 81 3B 27 1D C6 :09
---------------------------------------------------------
Sum: 03 AF 3F 73 4F 21 BA 5C 55 86 95 4C 56 01 DE C9 :A4
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5A00 06 17 0A 01 A6 A0 17 08 3F C6 04 17 09 F7 A6 A0 :F3
5A10 17 08 35 81 0D 27 12 C6 0F 17 09 E9 17 08 29 81 :C2
5A20 0D 27 06 31 21 A6 A4 20 F3 35 D0 34 30 10 9E 06 :06
5A30 0D 05 27 1A 1F 21 17 FE 31 26 0E 1F 31 C6 06 17 :40
5A40 00 DC 86 3A 17 00 E5 20 05 86 20 17 00 DE 5F A6 :5D
5A50 A4 81 11 26 04 C6 80 20 06 81 10 26 09 C6 40 31 :C3
5A60 21 0C 26 17 0F EA D7 25 A6 A4 17 0F E3 0C 26 30 :14
5A70 8D 15 8D 6D 84 27 34 E6 04 C4 C0 D1 25 26 28 E6 :13
5A80 04 C4 3C 27 04 A1 05 27 3C E6 04 C4 02 27 0C A1 :BC
5A90 06 10 27 01 8F A1 07 10 27 01 F3 E6 04 C4 01 27 :76
5AA0 06 A1 08 10 27 02 02 30 09 20 C8 30 8D 19 21 6D :6F
5AB0 84 27 2D E6 04 C4 C0 D1 25 26 06 A1 05 10 27 03 :48
5AC0 BC 30 06 20 EA E6 04 C4 20 10 26 00 7B E6 04 C4 :29
5AD0 10 10 26 00 BA E6 04 C4 08 10 26 01 BA 16 01 C0 :4E
5AE0 0D 05 27 2E 30 8D 05 D8 A6 80 81 20 27 06 81 3A :B0
5AF0 27 02 20 F4 6F 84 30 8D 05 45 9F 3B 30 8D 00 18 :E6
---------------------------------------------------------
Sum: 1D AC C1 11 72 4A 5F 5C 8B B9 23 47 B6 4E 3B 39 :38
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5B00 C6 05 8D 1A A6 9D 04 FE 17 0F 45 17 04 1F 86 0D :EF
5B10 8D 1A 86 01 97 26 35 B0 46 43 42 20 24 00 34 10 :23
5B20 A6 80 27 22 5D 27 1F 8D 03 5A 20 F4 34 10 30 8D :11
5B30 05 8E 6D 84 27 04 30 01 20 F8 81 08 27 06 A7 84 :D9
5B40 6F 01 35 90 6F 1F 35 90 0D 05 27 11 81 39 26 04 :B6
5B50 C6 01 D7 24 C6 04 8D C6 86 0D 17 FF CF 35 B0 0C :48
5B60 26 0D 05 27 F8 C6 04 17 FF B4 86 20 17 FF BD A6 :0A
5B70 21 17 0E DC A6 A4 81 1E 27 56 81 1F 27 52 30 8D :5E
5B80 19 2E 6D 84 27 42 A6 21 A4 02 26 04 30 06 20 F2 :80
5B90 E6 02 C1 FF 27 F6 C1 80 26 04 86 01 97 24 C1 40 :73
5BA0 27 0C C6 02 17 FF 77 86 2C 17 FF 80 20 DE 30 06 :04
5BB0 A6 A4 81 36 27 0B 81 37 27 07 86 55 17 FF 6D 20 :97
5BC0 E6 86 53 17 FF 66 20 DF 86 08 17 FF 5F 16 FF 88 :DA
5BD0 A6 21 84 80 27 0A A6 21 84 08 10 27 FF 02 20 08 :AF
5BE0 A6 21 84 08 10 26 FE F8 A6 21 44 44 44 44 C6 01 :1D
5BF0 30 8D 18 BC 6D 84 10 27 FE E6 84 0F A1 03 27 04 :FF
---------------------------------------------------------
Sum: A8 88 AE 8E C3 D7 02 44 04 FB 8D D5 52 5A DE 5E :95
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5C00 30 06 20 F0 81 0F 10 27 FE D6 5D 27 0F C6 02 17 :53
5C10 FF 0C 86 2C 17 FF 15 5F A6 21 20 D4 C6 02 17 FE :DF
5C20 FD 16 FF 34 0F 21 0F 27 C6 04 17 FE F1 86 20 17 :39
5C30 FE FA A6 21 17 0E 19 0C 26 0D 27 27 05 86 23 17 :4F
5C40 FE EA 0D 21 26 11 0D 27 26 2E 86 3C 17 FE DD E6 :6F
5C50 21 96 23 1F 01 20 09 A6 22 17 0D F4 0C 26 AE 21 :04
5C60 17 FC 2C 0D 05 26 02 35 B0 4D 26 0C C6 06 17 FE :BE
5C70 AD 86 0D 17 FE B6 35 B0 86 24 17 FE AF A6 21 17 :3C
5C80 02 AB 0D 21 27 05 A6 22 17 02 A2 16 FE CA 86 01 :EF
5C90 97 21 0F 27 16 FF 91 86 01 97 27 0F 21 16 FF 88 :A6
5CA0 86 01 97 27 97 21 16 FF 7F 0C 26 31 21 A6 A4 17 :76
5CB0 0D 9E C6 04 17 FE 67 86 20 17 FE 70 0F 28 0F 08 :6A
5CC0 A6 A4 84 80 27 66 A6 A4 84 10 27 09 86 01 97 28 :2F
5CD0 86 5B 17 FE 57 A6 A4 84 0F 81 04 27 76 81 00 10 :DD
5CE0 27 00 85 81 01 10 27 00 8B 81 02 10 27 00 89 81 :B4
5CF0 03 10 27 00 8D 81 08 10 27 00 8B 81 09 10 27 00 :D3
```

```
---------------------------------------------------------
Sum: 8F 9E 74 47 DF 0A C7 D0 0A 8C 30 E1 DE E4 9E C0 :2F
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5D00 9D 81 06 10 27 00 AD 81 05 10 27 00 AE 81 07 10 :0B
5D10 27 00 AC 81 0C 10 27 01 13 81 0D 10 27 01 33 A6 :4A
5D20 A4 84 9F 81 9F 10 27 01 3F 16 FD B4 E6 A4 4F C4 :C2
5D30 10 27 06 86 E0 AA A4 20 04 A6 A4 84 0F 17 01 A4 :AE
5D40 30 8D 03 2C C6 05 A6 80 5A 27 08 4D 27 F8 17 FD :E6
5D50 DB 20 F3 A6 A4 17 00 6B 0D 28 27 05 86 5D 17 FD :12
5D60 CB 86 0D 17 FD C6 35 B0 0D 28 10 26 FD 72 86 01 :7E
5D70 97 08 20 DF 86 02 20 F8 0D 28 10 26 FD 62 86 FF :8D
5D80 20 EE 86 FE 20 EA A6 21 17 0C C5 0C 26 0F 21 86 :33
5D90 3C 17 FD 98 4F E6 21 1F 01 17 00 6D 20 B5 0C 26 :E9
5DA0 A6 21 17 0C AB 0C 26 A6 22 17 0C A4 86 01 97 21 :95
5DB0 AE 21 20 E5 86 41 17 FD 73 20 98 86 42 20 F7 86 :3F
5DC0 44 20 F3 34 06 86 2C 17 FD 62 D6 08 10 2A 00 08 :D9
5DD0 86 2D 17 FD 57 5C 26 F8 35 06 84 60 26 04 86 58 :BF
5DE0 20 12 81 20 26 04 86 59 20 0A 81 60 27 04 86 55 :ED
5DF0 20 02 86 53 17 FD 35 34 04 D6 08 27 0A 2B 08 86 :44
---------------------------------------------------------
Sum: 9F 0F 45 8B D9 AE AB B5 DF 8E 70 78 E6 A8 93 A6 :81
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5E00 2B 17 FD 28 5A 26 F8 35 84 86 24 17 FD 1E 1F 10 :A3
5E10 0D 21 27 02 8D 05 1F 98 8D 01 39 34 06 17 06 1C :DA
5E20 96 0D 17 FD 07 96 0E 17 FD 02 35 86 0C 26 A6 21 :2C
5E30 17 0C 1D 86 3C 17 FC F4 E6 21 4F 1F 01 0F 21 17 :C6
5E40 00 72 30 8D 00 08 C6 03 17 FC D3 16 FF 0A 2C 50 :81
5E50 43 00 0C 26 A6 21 17 0B F7 0C 26 A6 22 17 0B F0 :61
5E60 86 01 97 21 AE 21 20 D7 86 01 97 21 0F 27 8D 08 :0F
5E70 86 08 17 FC B7 16 FE E0 34 30 16 FD B5 0C 26 C6 :70
5E80 04 17 FC 9A 86 20 17 FC A3 A6 21 17 0B C2 A6 04 :62
5E90 84 02 27 17 0C 26 A6 22 17 0B B5 86 01 97 21 AE :82
5EA0 21 17 00 10 86 0D 17 FC 83 35 B0 0F 21 E6 21 1D :AA
5EB0 1F 01 20 ED 34 10 9F 19 9E 06 D6 26 3A 1F 10 D3 :05
5EC0 19 1F 01 0D 10 27 04 86 01 97 1E 17 F9 C1 0D 05 :A0
5ED0 26 02 35 90 4D 26 07 C6 06 17 FC 42 35 90 35 10 :92
5EE0 17 FF 26 39 34 16 30 8D 01 86 6F 84 6F 01 6F 02 :D7
5EF0 81 80 25 05 C6 2D E7 84 40 80 64 25 0C C6 31 E7 :BC
---------------------------------------------------------
Sum: D3 9D 06 06 D8 2B B1 2D DF 83 D0 9E 05 34 B0 12 :28
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5F00 01 C6 30 E7 02 E7 03 20 02 8B 64 C6 30 80 0A 25 :80
5F10 03 5C 20 F9 8B 0A C1 30 27 02 E7 02 8A 30 A7 03 :74
5F20 35 96 86 20 17 03 21 86 20 17 03 1C 39 17 05 0C :E9
5F30 96 0D 17 FB F7 96 0E 17 FB F2 39 00 00 00 00 00 :8D
5F40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5F50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5F60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5F70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5F80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5F90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5FA0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5FB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5FC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5FD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5FE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5FF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
---------------------------------------------------------
Sum: CF C5 ED FB 9B 8A F3 ED 44 96 87 E4 F3 C7 B6 34 :6A
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6000 1C 00 16 01 16 32 62 84 00 10 00 01 01 30 31 00 :D4
6010 00 00 26 62 82 62 48 00 0B FD C4 96 00 00 00 00 :16
6020 00 01 06 00 00 00 00 00 01 00 08 14 00 60 34 00 :B8
6030 01 00 00 0D 0A 60 68 75 00 10 00 60 3F 60 3F 35 :D8
6040 D6 FB FA 00 00 00 00 00 00 00 00 00 00 00 00 00 :CB
6050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6060 00 00 00 00 00 00 00 00 4C 46 42 46 41 00 00 00 :5B
6070 00 00 00 34 00 00 00 00 00 00 00 00 00 00 00 00 :34
6080 42 0D 30 0D 38 2C 36 32 38 32 0D 0D 43 0D 0D 0D :46
6090 0D 0D 50 43 0D 00 00 00 00 00 00 00 00 00 00 00 :BA
60A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 80 :80
60B0 01 56 60 4C F5 60 CA 4F 31 01 56 60 4C F5 60 CA :C4
60C0 00 0D 55 4C 53 20 41 2C 42 2C 58 2C 55 2C 50 43 :94
60D0 0D 00 0D 00 00 00 00 00 00 00 00 00 00 00 00 00 :1A
60E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
60F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
---------------------------------------------------------
Sum: 50 79 7E 8C 2F A0 53 A6 03 C2 C9 EA 65 1E 61 CF :C6
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6100 16 02 05 16 03 6C 16 01 3F 16 02 2F 16 03 BC 16 :2A
6110 03 75 16 02 CD 16 03 24 16 03 00 30 8D FE E6 1F :73
6120 10 1F BB 0F 05 0F 1D 0F 10 0F 1F 20 70 0F 05 0F :FA
6130 1D 0F 1F 86 0D 17 01 10 86 3E 17 01 0B 17 00 E0 :E4
6140 A6 A4 81 4C 10 27 01 CD 81 44 10 27 05 23 81 49 :0A
6150 10 27 03 C8 81 32 10 27 06 29 81 31 10 27 06 20 :2A
6160 81 50 27 23 81 42 27 16 81 53 27 15 81 52 27 2D :52
6170 81 54 10 27 00 1D 81 4D 10 27 04 FE 20 AF 7E AF :2C
6180 A2 9E 39 6F 84 20 A6 96 0F 27 04 0F 0F 20 0E 03 :51
6190 0F 20 0A 0D 10 27 04 0F 10 20 02 03 10 86 0D 17 :7F
61A0 00 A6 17 01 3C 46 4D 2D 38 20 36 38 30 39 20 41 :4A
61B0 53 53 45 4D 42 4C 45 52 0D 54 45 58 54 3A 00 17 :00
```

```
61C0 03 33 9E 13 17 02 1B 17 01 17 20 4C 41 42 45 4C :CA
61D0 3A 00 9E 37 6D 84 27 04 30 08 20 F8 17 02 03 17 :AE
61E0 00 FF 20 50 52 49 4E 54 45 52 3A 00 96 0F 27 08 :51
61F0 17 00 EE 4F 4E 00 20 07 17 00 E6 4F 46 46 00 17 :B8
---------------------------------------------------------
Sum: 56 FD 69 BE 2A 08 DC 35 F4 79 D5 20 AB 24 7D 5D :C8


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6200 00 DF 0D 4C 49 53 54 3A 00 0D 10 27 0A 17 00 D1 :98
6210 4F 46 46 00 16 FF 16 17 00 C7 4F 4E 00 16 FF 0D :A3
6220 34 56 30 8D FE 5A 86 5F 8D 1E 86 08 8D 1A 8D 54 :45
6230 27 FC 81 0D 27 6A 81 08 27 76 81 1D 27 72 81 20 :40
6240 25 E4 8D 04 A7 80 20 DE 81 0D 26 09 8D 07 86 0A :A0
6250 8D 03 86 0D 39 34 02 96 1D 35 02 27 02 8D 64 0D :A3
6260 1F 26 02 8D 01 39 34 56 30 8D FD BF 33 8D FD C4 :92
6270 A7 C4 CC 14 00 ED 84 EF 02 CC 00 01 ED 04 AD FF :17
6280 FB FA 35 D6 34 30 31 8D FD A9 6F A4 30 8D FD 9B :30
6290 CC 15 00 ED 84 10 AF 02 AD FF FB FA A6 A4 35 B0 :E3
62A0 86 20 8D A4 86 0D A7 80 8D 9E 31 8D FD D2 35 D6 :54
62B0 6D 1F 27 0C 86 20 8D 90 86 08 8D 8C 8D 8A 6F 82 :31
62C0 16 FF 63 34 56 30 8D FD 62 33 8D FD 67 A7 C4 CC :79
62D0 0E 00 ED 84 EF 02 CC 00 01 ED 04 AD FF FB FA 35 :04
62E0 D6 AE E1 A6 80 27 05 17 FF 5E 20 F7 1F 15 8D F1 :F4
62F0 43 4D 44 45 52 52 0D 00 16 FE 32 8D E4 46 4D 54 :68
---------------------------------------------------------
Sum: 19 90 43 AE 40 08 CA 24 B9 CD 96 6F 36 68 0F 15 :1D


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6300 45 52 52 0D 00 86 01 39 8D D7 50 47 4D 45 52 52 :E7
6310 0D 00 86 01 39 A6 21 81 0D 27 0E 17 01 67 26 CE :CA
6320 8C 00 00 27 07 30 1F 20 03 8E 00 00 9F 11 17 01 :82
6330 CD 26 BB 86 0D 17 FF 10 10 9E 15 A6 A4 10 27 FD :A8
6340 EC 8D 70 17 03 A5 30 01 17 00 97 86 20 17 FE F8 :3A
6350 17 04 36 A6 A4 81 3B 27 49 C6 06 17 00 A7 A6 A0 :97
6360 81 3A 27 06 81 3D 27 33 86 20 17 FE DB C6 04 17 :77
6370 00 93 81 3B 10 27 00 EE 17 FE CD 31 21 81 0D 27 :5D
6380 2D C6 0E A6 A4 81 2F 27 19 17 00 79 31 21 81 0D :AB
6390 27 09 34 02 86 20 17 FE AF 35 02 17 FE AA 81 0D :54
63A0 27 0C C6 1E 17 00 22 31 21 86 0D 17 FE 9A 17 03 :FE
63B0 DB 20 88 17 FE CE 81 20 26 0E 17 FE C7 27 FB 81 :BA
63C0 20 27 05 35 06 16 FD 65 39 A6 A4 81 0D 27 12 17 :60
63D0 FE 76 31 21 5A 26 F2 A6 A4 81 0D 27 04 31 21 20 :AD
63E0 F6 39 34 04 1F 10 17 00 53 96 0D 17 FE 5A 96 0E :B6
63F0 17 FE 55 1F 98 17 00 44 96 0D 17 FE 4B 96 0E 17 :3A
---------------------------------------------------------
Sum: B0 A5 30 0F DB CF C1 F8 7F B8 EF 32 FB A6 56 EE :34


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6400 FE 46 35 04 39 A6 A4 81 27 27 20 17 00 64 27 0B :9C
6410 17 FE 35 31 21 5A 2A ED 5C 20 EA 5D 27 0C 34 02 :39
6420 86 20 17 FE 23 35 02 5A 20 F1 39 17 FE 1A 5A 2A :6C
6430 01 5C 31 21 A6 A4 81 0D 27 E1 20 D4 34 06 1F 89 :65
6440 84 0F 81 0A 24 14 8A 30 97 0E 54 54 54 54 C4 0F :D8
6450 C1 0A 24 0C CA 30 D7 0D 35 86 80 09 8A 40 20 E8 :EF
6460 C0 09 CA 40 20 F0 34 02 C6 10 17 FF AE 35 02 16 :00
6470 FF 30 81 20 27 0E 81 0D 27 0A 81 3A 27 06 81 3D :6A
6480 27 02 81 3B 39 34 20 31 21 8E 00 00 86 01 97 0C :7C
6490 A6 A0 80 30 25 54 81 0A 25 1C 80 07 81 10 24 4A :C1
64A0 20 14 A6 A0 8D 25 27 47 80 30 25 3E 81 0A 25 06 :63
64B0 80 07 81 10 24 34 34 06 1F 10 58 49 58 49 58 49 :BC
64C0 58 49 1F 01 35 06 1E 89 3A 20 D7 81 3B 27 1A 81 :52
64D0 20 27 16 81 2C 27 12 81 5D 27 0E 81 3A 27 0A 81 :C3
64E0 0D 27 06 81 2B 27 02 81 2D 39 35 20 86 01 39 35 :40
64F0 06 31 3F 4F 39 9E 39 A6 80 26 FC 9F 13 39 8E 00 :96
---------------------------------------------------------
Sum: 98 97 44 37 2C EE CE DA AC 57 E2 44 FA 4B 5E E6 :1E


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6500 00 DE 39 9C 11 27 0E A6 C0 27 0E 81 0D 27 02 20 :6B
6510 F6 30 01 20 EE DF 15 4F 39 86 01 39 17 FF 66 26 :13
6520 3F 9F 11 17 00 7B 26 FB 17 02 5B 26 F6 17 FF C5 :0A
6530 17 FF CB 26 2B 17 00 2B DC 15 1F 01 DC 13 93 15 :1C
6540 1F 02 DC 15 D3 17 1F 03 17 00 2C 30 8D FB 71 10 :9A
6550 9E 17 DE 15 17 00 20 DC 11 C3 00 01 DD 11 20 C3 :61
6560 16 FD 8B 30 8D FB 59 CE 00 01 A6 80 81 0D 27 04 :5D
6570 33 41 20 F6 DF 17 39 DF 19 9C 19 24 18 10 9F 19 :6A
6580 1F 10 D3 19 1F 01 1F 30 D3 19 1F 03 A6 82 A7 C2 :29
6590 31 3F 26 F8 39 A6 80 A7 C0 31 3F 26 F8 39 30 8D :D8
65A0 FB 1E 17 FC 7B 17 01 38 81 0D 10 27 00 A7 81 3B :1F
65B0 10 27 00 8C 34 20 A6 A0 17 FE B7 26 F9 35 20 81 :1E
65C0 3A 27 0A 81 3D 27 06 86 20 A7 80 20 2B C6 07 A6 :E1
65D0 A4 A7 80 81 3A 27 18 81 3D 27 07 31 21 5A 27 7A :FE
65E0 20 ED C1 06 24 74 31 21 17 00 F5 C6 1F 20 37 C1 :C7
65F0 06 24 67 31 21 17 00 E8 C6 05 A6 A4 17 FE 73 27 :A6
```

```
---------------------------------------------------------
Sum: B1 76 3D 1B 43 75 AF 66 92 4C BB E7 12 4E A1 23 :F0


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6600 0B A7 84 31 21 30 01 5A 27 50 20 EE 81 04 25 4A :8C
6610 17 00 CD 81 0D 27 35 81 3B 27 25 A6 A4 17 FE 52 :87
6620 27 38 86 20 A7 80 A6 A4 81 2F 27 31 17 FE 43 27 :FD
6630 08 A7 84 31 21 30 01 20 ED 17 00 A4 81 0D 27 0C :3F
6640 86 3B A7 80 31 21 A6 A4 81 0D 26 F6 86 0D A7 80 :E8
6650 31 21 86 00 39 35 03 16 FA D3 16 FC 9E A7 80 31 :34
6660 21 A6 A4 81 0D 27 F3 81 2F 26 F2 A7 80 31 21 20 :74
6670 C8 8D 1F 10 26 FC 77 16 FA B3 34 20 8D 14 35 20 :2A
6680 10 26 FC 6A 17 FD FE 10 26 FC 63 30 1F 9F 11 16 :58
6690 FE 91 17 FD F0 26 46 34 10 30 1F 9F 11 17 FE 5E :B5
66A0 35 10 26 39 DE 15 A6 A4 81 0D 27 05 17 FD D6 26 :AB
66B0 2C 9F 11 34 40 17 FE 46 35 40 26 21 1F 30 93 15 :5E
66C0 2A 1B 9E 15 34 10 17 FE 2C 35 10 10 9E 13 9F 13 :35
66D0 1F 20 93 13 27 07 1F 02 17 FE 9C 4F 39 86 01 39 :2D
66E0 A6 A4 81 20 26 04 31 21 20 F6 39 96 05 0F 1F 81 :00
66F0 32 27 01 39 34 30 0D 10 27 04 86 01 97 1F 17 00 :93
---------------------------------------------------------
Sum: 81 81 48 69 6D 1A 4C 4F EA 1C 08 0D C7 C9 58 3C :14


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6700 8E 10 26 00 08 0D 28 27 02 0F 1F 35 B0 0F 1F 35 :A0
6710 30 35 06 0D 10 10 27 FA 14 30 01 17 FC C4 17 FB :E7
6720 E7 16 FA 09 00 00 00 00 00 00 00 00 00 00 00 00 :00
6730 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6740 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6750 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6760 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6770 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
---------------------------------------------------------
Sum: A5 5B 26 16 18 1D 4F 21 16 3F 20 4C AC D3 36 30 :87
```

FM-8

6809 デバッガ

ETF Laboratory ■笠作貴弥

ＭＺのＳ．Ｒ．Ｄを目標にこのプログラムを制作しました．

プログラムの特徴

ダイレクト・ページ・レジスタの活用により，256バイト単位でのリロケータブルとなっています．また，ハードウェアとのインターフェイスはすべてＢＩＯＳを通じて行なっているので，メモリ上にはＢＩＯＳのみあればＯＫです．ただし，デバッガ自身を起動するプログラムは必要ですが…….

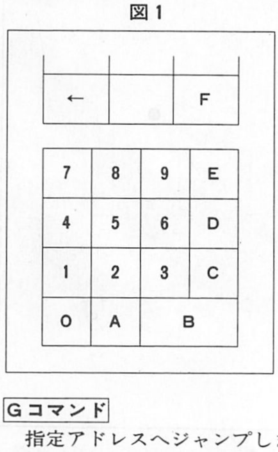
コマンドの説明

Hコマンド

コマンド・リストを表示します．

Mコマンド

図１



このコマンド使用中に限り，テン・キー部分が16進キーになります(図１)．また，16バイト入力ごとにベルが鳴り，チェック・サムを表示します．アドレスの変更は[←]でバック，[SP]でインクリメントします．他のキー入力でコマンド待ちになります．

Gコマンド

指定アドレスへジャンプします．ユーザープログラムの最後にＲＴＳがあるとF-BASICへ戻ります．

Cコマンド

レジスタ・バッファの値（ＰＣとＳＰは除く）を各レジスタに入れてコールします．実行後はレジスタの値をバッファにストアして，コマンド待ちになります．

Sコマンド

セーブのコマンドです．F-BASIC と同じフォーマットで記録します．

Lコマンド

ロードのコマンドです．アドレスを指定すると，そのアドレスからロードを開始します．

Vコマンド

カセットとメモリの内容を照合します．

Wコマンド

画面の横文字数を変更します．

Iコマンド

指定アドレスから入力された文字列をキャラクタ・コードにしてストアします．

Pコマンド

指定されたメモリ領域を同一のデータで埋めます．

Xコマンド

ブロック転送のコマンドです．デバッガ自身も転送できます．

Rコマンド

レジスタ・バッファの内容を表示します．カーソルを移動してエディットができます．

@コマンド

F-BASICのMotor ONと同様です．

!コマンド

F-BAISCへリターンします．

?コマンド

ＣＲＴ画面のハード・コピーをします．

Zコマンド

メモリ・ブロックの比較コマンドです．

Aコマンド

ディスアセンブルのコマンドです．プリンタへも出力できます．

Dコマンド

D [RET] で以前にダンプした次のアドレスから64バイト分メモリ・ダンプします．

D（アドレス）[RET] で，指定アドレスから64バイト分ダンプします．

D先頭アドレス，最終アドレスで指定アドレス間をダンプします．それからオプションAでASCII，Cで縦横チェック・サム方式でダンプします．また，一度画面に表示されたものはカーソル・エディットできます．一度に複数行の変更が可能です．

各コマンドの注意・その他

❶S, L, Vコマンドおよびプリンタ出力中にSTOPを押すと, F-BASICへ戻ってしまいます.
❷D, A, Zコマンドについては ESC で一時停止 CTRL C で中断します.
❸I, P, Xコマンドについてはコマンド実行前に〝Are you sure ?〞と確認してくるので, Y または N を入力してください.

プログラムについて

前にも書いたように256バイト単位でのリロケータブルとなっているので××00からロードして使えます. また, ワーク・エリアとして, プログラムの先頭より前に256バイト必要なので注意してください. たとえばプログラムを＄1000からロードした場合, ＄0F00～＄0FFFがワーク・エリアとなります.

それから, スタックはF-BASIC でセットされたエリアを使うので, 必要に応じてCLEAR文で変更してください.

最後に

カセット・ベースの小さなアセンブラを使ったため, 何んと7回に分けてアセンブルしました. ですから, プログラム中にかなりゴミがあります. 大きなアセンブラを使って再アセンブルすれば, かなり短縮できると思います.

また, 初めミニアセンブルやトレーサも内蔵させる予定だったのですが, まだ完全ではないので, 今回はカットしました. 数バイトの変更でそれらをリンクできるようになっているので, 完成したら紹介させていただきます.

機能的にはまだ充分ではありませんが, F-BASIC のモニタよりは役立つはずです. ぜひ使ってみてください.

コマンド表

コマンド		書式, 機能
H	Help	コマンド・リストの表示
M	メモリ・チェンジ	M アドレス RET
S	セーブ	S 先頭アドレス, 最終アドレス, 実行開始アドレス RET
L	ロード	L RET L ロード開始アドレス RET
V	ベリファイ	カセットとメモリの照合
@	モータON	F-BASICのMOTOR ONと同じ
!	クィット	F-BASICへリターン
?	ハード・コピー	画面のハード・コピー
W	ワイス・チェンジ	画面の横文字数の変更
G	ジャンプ	G アドレス RET
C	コール	C アドレス RET
R		スクリーン・エディット可能
X	トランスファ	X 先頭アドレス, 最終アドレス, 転送先アドレス RET
P	クリア	P 先頭アドレス, 最終アドレス, クリア・データ RET
I	文字列書き込み	I 先頭アドレス, 文字列 RET
Z	メモリ・ブロック比較	Z 先頭アドレス, 最終アドレス, 比較先先頭アドレス RET (, P)
A A	ディスアセンブル	A 先頭アドレス, 最終アドレス(.P) , Pでプリンタ出力 RET
D	メモリ・ダンプ	D RET D 先頭アドレス, 最終アドレス RET (,, P) D 先頭アドレス, 最終アドレス, オプション (, P) RET オプションA：ASCII ダンプ C：チェック・サム AC：ASCII&チェックサム

コマンド・リスト

```
6809 Super DEBUGGER FM-8 Version 1.0
  Copyright (C) 1982 By HIRO/ETF Lab.
H: Help
M: Memory Change
G: Jump (User Prog.)
C: Call (User Prog.)
S: Save
L: Load
V: Verify
W: Width Change
I: ASCII Write
P: Memory Paint
X: Transfer
A: Mini Assembler
D: Memory Dump
R: Register Display
Z: Memory Compare
@: Motor ON
!: Quit
?: CRT Copy
:?
```

各コマンドの実行例

```
:R
* A-00  B-00 CC-00 DP-55  X-1111  Y-AA00  U-0000  S-0000 PC-1234

:P 1000,2000,0
Are you sure ? (Y/N) Y
:I 1000,I/O I/O
Are you sure ? (Y/N) Y
:D 1000,1010,AC
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1000 49 2F 4F 20 49 2F 4F 20 00 00 00 00 00 00 00 00 :CE  I/O I/O ........
1010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00  ................
------------------------------------------------------------
Sum: 49 2F 4F 20 49 2F 4F 20 00 00 00 00 00 00 00 00 :CE


:X 5000,5500,1000
Are you sure ? (Y/N) Y
:
:
:?
```

6809デバッガ FM-8 Version1.0 ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6800 34 08 1F 50 4A 1F 8B 16 03 4B 9F 52 DD 54 86 14 :BF
6810 97 50 30 8D FF 3A 7F 03 13 6E 9F FB FA 97 58 86 :E9
6820 0E 20 04 97 58 86 14 97 50 30 8D FF 2B 9F 52 8E :08
6830 00 01 9F 54 20 DC 34 16 D6 31 27 08 53 27 09 17 :0A
6840 04 78 35 96 8D DD 35 96 8D D3 35 96 86 0D 8D E6 :AD
6850 86 0A 20 E2 96 31 43 27 08 86 11 8D D9 86 07 8D :E2
6860 D5 39 86 20 20 D0 34 12 86 17 20 04 34 12 86 16 :8D
6870 97 59 86 10 97 50 30 8D FE DC 9F 52 8E 00 04 9F :26
6880 54 0F 56 0F 57 86 0C 97 58 17 FF 86 35 92 8D D6 :66
6890 8D 0D 20 D8 34 14 C6 13 D7 50 8E 00 00 20 15 34 :D1
```

```
68A0 14 C6 12 D7 50 8E 0D 0A 9F AA 8E 11 07 9F AC 97 :89
68B0 AE 8E 00 05 9F 54 30 8D FE F0 9F 52 8E 00 55 9F :52
68C0 56 17 FF 4E 30 8D FE E2 D6 55 3A 6F 84 35 94 8D :05
68D0 95 8D 05 4D 27 FB 20 94 34 10 86 15 97 50 30 8D :CD
68E0 FE C8 9F 52 8E 00 01 9F 54 17 FF 26 96 AA 35 90 :7A
68F0 34 12 30 8D 02 12 CC 00 17 17 FF 0E 8D D1 17 FF :92
------------------------------------------------------------
Sum: 8F 7B AE AD FC FF 28 78 96 FA 6F 6E 7E A7 0A 50 :EC
```

6809デバッガ　FM-8 Version1.0　ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6900 35 81 59 35 92 8D 02 1F 98 34 02 47 47 47 47 8D :FB
6910 02 35 02 84 0F 81 0A 25 02 8B 07 8B 30 16 FF 16 :F6
6920 86 20 A1 80 27 FC 30 1F 39 32 7E 86 07 17 FE F3 :B7
6930 32 62 39 8D EB A6 80 27 F2 81 2C 26 EE 39 30 8D :3B
6940 FE 6E 8D DC A6 84 27 E3 8D 2C 24 DF 34 06 8D D0 :5C
6950 0F 32 0F 33 86 04 97 34 A6 84 8D 1A 24 08 8D 08 :6A
6960 30 01 0A 34 26 F2 35 86 C6 04 08 33 09 32 5A 26 :02
6970 F9 9A 33 97 33 39 81 30 25 10 80 30 81 0A 25 0B :1A
6980 81 11 25 06 80 07 81 10 25 01 4F 39 30 8D 00 0E :4E
6990 9F 52 86 10 97 50 8E 00 0B 9F 54 16 FE 74 00 00 :82
69A0 01 00 28 19 00 19 00 01 00 E0 20 36 38 30 39 20 :53
69B0 53 75 70 65 72 20 44 45 42 55 47 47 45 52 20 46 :DA
69C0 4D 2D 38 20 56 65 72 73 69 6F 6E 20 31 2E 30 20 :87
69D0 E0 20 20 20 43 6F 70 79 72 69 67 68 74 20 28 43 :84
69E0 29 20 31 39 38 32 20 42 79 20 48 49 52 4F 2F 45 :BE
69F0 54 46 20 4C 61 62 2E 0D 0A 48 3A 20 48 65 6C 70 :39
---------------------------------------------------------
Sum: 43 FE FA F9 F3 5B B3 E8 B3 4B 4D 97 38 7C 59 B8 :C4

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6A00 0D 0A 4D 3A 20 4D 65 6D 6F 72 79 20 43 68 61 6E :D1
6A10 67 65 0D 0A 47 3A 20 4A 75 6D 70 20 28 55 73 65 :95
6A20 72 20 50 72 6F 67 2E 29 0D 0A 43 3A 20 43 61 6C :45
6A30 6C 20 28 55 73 65 72 20 50 72 6F 67 2E 29 0D 0A :79
6A40 53 3A 20 53 61 76 65 0D 0A 4C 3A 20 4C 6F 61 64 :79
6A50 0D 0A 56 3A 20 56 65 72 69 66 79 0D 0A 57 3A 20 :04
6A60 57 69 64 74 68 20 43 68 61 6E 67 65 0D 0A 49 3A :00
6A70 20 41 53 43 49 49 20 57 72 69 74 65 0D 0A 50 3A :55
6A80 20 4D 65 6D 6F 72 79 20 50 61 69 6E 74 0D 0A 58 :24
6A90 3A 20 54 72 61 6E 73 66 65 72 0D 0A 41 3A 20 44 :95
6AA0 69 73 2D 41 73 73 65 6D 62 6C 65 72 20 0D 0A 44 :22
6AB0 3A 20 4D 65 6D 6F 72 79 20 44 75 6D 70 0D 0A 52 :F2
6AC0 3A 20 52 65 67 69 73 74 65 72 20 44 69 73 70 6C :BB
6AD0 61 79 0D 0A 5A 3A 20 4D 65 6D 6F 72 79 20 43 6F :F0
6AE0 6D 70 61 72 65 0D 0A 40 3A 20 4D 6F 74 6F 72 20 :F7
6AF0 4F 4E 0D 0A 21 3A 20 51 75 69 74 0D 0A 3F 3A 20 :82
---------------------------------------------------------
Sum: 7D F4 FF BF 72 34 D2 FC 37 CF C9 61 CE A5 13 8E :E7

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6B00 43 52 54 20 43 6F 70 79 0D 0A 41 72 65 20 79 6F :DB
6B10 75 20 73 75 72 65 20 3F 20 28 59 2F 4E 29 20 48 :62
6B20 00 06 5A 07 B1 40 05 F8 21 00 10 57 00 14 43 02 :36
6B30 C5 50 00 28 49 00 63 4D 00 7C 47 01 28 44 02 F7 :5F
6B40 53 06 95 4C 05 2D 56 05 27 58 02 09 41 08 4C 52 :38
6B50 02 73 3F 01 BD 17 FE 34 30 8D FE 4D CC 01 5F 17 :06
6B60 FC A8 0F 31 0F AF 86 3A 17 FD 23 96 AE 81 3A 10 :A8
6B70 26 01 6A 30 8C A9 C6 12 96 AF A1 80 27 07 30 02 :94
6B80 5A 26 F7 20 DD 1F 52 EC 84 AD AB 20 D5 30 8D FE :5D
6B90 66 CC 01 11 16 FC 73 32 62 35 88 A6 8D FE 03 81 :CF
6BA0 28 26 03 48 20 02 86 28 A7 8D FD F6 16 FD DD 17 :97
6BB0 FD 8C DC 32 DD 3A 17 FD 7A 17 FD 86 DC 32 DD 3C :FD
6BC0 17 FD 70 17 FD 7C DC 3C 9E 3A 93 3A 10 25 FD 59 :5C
6BD0 1F 02 31 21 0D 32 10 26 FD 4F 17 FD 13 10 26 FD :8E
6BE0 48 96 33 A7 80 31 3F 26 FA 39 17 FD 51 17 FD 43 :BD
6BF0 17 FC FD 10 26 FD 32 10 9E 32 A6 80 27 04 A7 A0 :ED
---------------------------------------------------------
Sum: 6E 1F 16 0C AC E3 57 5D 8C B9 49 5B AC DF 04 36 :A0

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6C00 20 FB 39 0F 31 A6 8D FD 99 81 28 27 04 86 07 20 :DB
6C10 02 86 03 97 36 17 FD 26 9E 32 17 FC 2F 1F 10 17 :EA
6C20 FC E3 17 FC 3D A6 84 17 FC DF 86 2D 17 FC 07 17 :2F
6C30 FC 9D 8D 34 48 48 48 48 97 35 17 FC 92 8D 29 9A :3B
6C40 35 A7 80 1F 10 C4 0F 27 06 D4 36 26 D5 20 CB 86 :01
6C50 07 17 FB E2 86 3A 17 FB DD 30 10 4F C6 10 AB 80 :3A
6C60 5A 26 FB 17 FC A3 20 B2 1F 89 17 FD 09 24 08 1F :13
6C70 89 17 FC 9F 1F 98 39 86 0A C1 2E 27 24 C1 0D 27 :EA
6C80 1F C1 2D 27 1A C1 2B 27 15 C1 2C 27 10 C1 1C 27 :9E
6C90 0B C1 20 27 0E C1 1D 27 10 16 FC 8F 4C 4C 4C 4C :07
6CA0 4C 20 CC 30 01 32 62 20 9A 30 1F 32 62 20 9E 17 :6F
6CB0 FC 8C 32 62 35 08 6E 9D FA 78 34 10 9E 38 A7 80 :17
6CC0 9F 38 35 90 34 16 30 8D FA 90 9F 52 DC 38 9F 38 :09
6CD0 93 52 DD 54 86 14 97 50 17 FB 37 35 96 81 2A 10 :66
6CE0 27 04 F2 30 8D FA C7 A6 8D FC B7 81 28 27 03 30 :84
6CF0 88 18 30 88 1C 9F 3C 0D AB 10 27 FE 65 30 8D FA :58
---------------------------------------------------------
Sum: 8C CD D1 09 5E 63 B7 77 D8 2B 96 E3 FF B8 D8 B0 :DD

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6D00 AD A6 84 17 FC 70 24 12 17 FC 37 0D 34 26 0B 10 :5C
6D10 9E 32 A6 80 27 04 81 20 27 06 17 FC 0C 16 FE 42 :64
6D20 A6 84 17 FC 51 24 F3 A6 01 17 FC 4A 24 EC A6 02 :61
6D30 81 20 26 E6 17 FC 0B 96 33 A7 A0 9C 3C 25 D3 17 :C2
6D40 FB 52 20 B3 34 10 03 31 5F 4F 34 06 DD AD DD AF :96
6D50 86 06 97 AC 86 11 97 50 8E 00 07 9F 54 30 8D FA :8C
6D60 49 9F 52 8E 00 0A 9F 56 17 FA A7 96 AE 26 02 86 :71
6D70 20 17 FA C2 35 06 4C A1 8D FC 27 26 CD 17 FA CC :9B
6D80 17 FB 55 81 03 27 07 5C E1 8D FC 17 26 BB 35 90 :9C
6D90 17 FB AB DC 32 DD 3A 17 FB 99 17 FB A5 DC 32 DD :2F
6DA0 3C 10 93 3A 10 25 FB 81 17 FB 88 17 FB 94 DC 32 :18
6DB0 DD 3E 17 FB 3B 10 26 FB 70 34 36 9E 3A 9C 3E 25 :4A
6DC0 0F 10 9E 3E A6 80 A7 A0 9C 3C 25 F8 27 F6 35 B6 :65
6DD0 DC 3C 93 3A D3 3E 1F 02 9E 3C A6 84 A7 A4 30 1F :B5
6DE0 31 3F 9C 3A 24 F4 35 B6 20 41 20 42 43 43 44 50 :26
6DF0 20 58 20 59 20 55 20 53 50 43 17 FA 4F 86 01 97 :EA
---------------------------------------------------------
Sum: DF B1 01 C5 B7 05 A5 80 10 56 C6 CF AC 91 13 E6 :68

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6E00 31 30 8D F9 55 9F 38 17 FA 4A 86 2A 17 FA 27 31 :87
6E10 8D F9 2F 30 8C D2 C6 04 8D 23 A6 A0 17 FA EA 17 :15
6E20 FA 40 5A 26 F3 86 05 97 35 8D 12 EC A1 17 FA D5 :16
6E30 17 FA 2F 0A 35 26 F2 17 FA 12 16 FE 87 A6 80 17 :92
6E40 F9 F4 A6 80 17 F9 EF 86 2D 16 F9 EA 17 FA EF 34 :F2
6E50 09 9E 46 10 9E 48 DE 4A 96 44 1F 8A 96 45 1F 8B :13
6E60 EC 8D F8 DE AD 9D F8 CA ED 8D F8 D6 1F B8 97 45 :56
6E70 1F A8 97 44 35 09 9F 46 10 9F 48 DF 4A 39 7F 04 :A1
6E80 3D 7F 04 3E 30 8D F8 D2 9F 38 17 F9 BF 86 01 97 :49
6E90 31 30 8D F9 1B 17 FA 88 A6 84 26 0D 9E 40 9F 3A :AF
6EA0 DC 3A C3 00 3F DD 3C 20 5F 17 FA 96 DC 32 DD 3A :7C
6EB0 17 FA 6D A6 84 27 E9 17 FA 79 17 FA 85 DC 32 DD :C3
6EC0 3C 17 FA 5C A6 84 27 40 17 FA 68 17 FA 52 A6 80 :3C
6ED0 81 41 26 0A 73 04 3D 17 FA 46 A6 80 27 2A 81 43 :38
6EE0 26 1E 73 04 3E 17 FA 38 A6 84 27 1C 17 FA 44 17 :1B
6EF0 FA 2E A6 84 81 50 10 26 FA 2F 86 FF 97 31 20 08 :F7
---------------------------------------------------------
Sum: 1A B1 BA D6 86 9B DE EF C5 D1 B5 25 F9 5C E9 06 :FD

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6F00 81 2C 10 26 FA 23 20 E7 9E 3A 96 31 43 27 12 A6 :C8
6F10 8D FA 8F 81 28 26 0A 7F 04 3D 7F 04 3E 86 08 20 :1E
6F20 02 86 10 97 35 86 10 97 36 CE 04 3F C6 11 6F C0 :DE
6F30 5A 26 FB 7D 04 3E 27 0D 17 F9 19 31 8D 01 2E 17 :9B
6F40 00 EA 17 FD 7F 17 F9 0C 1F 10 17 F9 B8 17 F9 12 :B2
6F50 0F 37 CE 04 3F D6 35 A6 84 9B 37 97 37 A6 84 AB :01
6F60 C4 A7 C0 A6 84 BB 04 4F B7 04 4F A6 80 17 F9 99 :3C
6F70 17 F8 EF 5A 26 E1 86 3A 17 F8 BB 96 37 17 F9 89 :4F
6F80 7D 04 3D 27 2D 17 F8 DA 17 F8 D7 C6 10 30 10 A6 :9D
6F90 80 81 20 25 0A 81 7F 27 06 81 F8 24 06 20 0D 86 :D3
6FA0 2E 20 09 34 02 96 31 43 35 02 27 F3 17 F8 87 5A :D8
6FB0 26 DD 17 F8 97 17 FD 0C 9C 3C 27 02 24 19 17 F9 :17
6FC0 17 81 03 27 19 7D 04 3E 10 27 FF 79 0A 36 10 26 :BF
6FD0 FF 73 8D 0D 16 FF 4E 7D 04 3E 27 02 8D 03 9F 40 :C6
6FE0 39 17 F8 70 C6 39 86 2D 17 F8 4B 5A 26 F8 17 F8 :4B
6FF0 5B 17 FC D0 17 F8 5D 31 8D 00 AD 8D 2F CE 04 3F :E2
---------------------------------------------------------
Sum: 4F 36 3F A8 9F 88 F3 AE 06 F9 C5 B2 B7 0A AB 98 :AE

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7000 C6 10 A6 C0 17 F9 02 17 F8 58 5A 26 F5 86 3A 17 :01
7010 F8 24 A6 C4 17 F8 F2 17 F8 32 17 F8 2F 17 FC A4 :BD
7020 96 31 43 26 06 17 F8 24 17 F8 21 39 A6 A0 27 FB :3A
7030 17 F8 03 20 F7 0D 0A 46 6F 75 6E 64 3A 20 20 13 :C9
7040 05 2D 20 4E 6F 20 45 72 72 6F 72 73 20 44 65 76 :EB
7050 69 63 65 20 49 2F 4F 20 45 72 72 6F 72 07 20 42 :AB
7060 61 64 20 46 69 6C 65 20 4D 6F 64 65 07 41 64 64 :1A
7070 20 20 2B 30 20 2B 31 20 2B 32 20 2B 33 20 2B 34 :91
7080 20 2B 35 20 2B 36 20 2B 37 20 2B 38 20 2B 39 20 :AA
7090 2B 41 20 2B 42 20 2B 43 20 2B 44 20 2B 45 20 2B :F1
70A0 46 20 53 75 6D 0D 0A 00 53 75 6D 3A 20 00 0F 35 :85
70B0 0F 3E 20 18 0F 35 86 01 97 3E 30 8D F6 F2 17 F8 :D9
70C0 5F A6 84 27 07 17 F8 7A 03 35 DE 32 17 00 B0 17 :66
70D0 00 B1 10 26 00 EB 8E 04 3D A6 81 26 F2 A6 08 81 :0F
70E0 02 10 26 00 E6 34 10 30 8D FF 4A CC 00 09 17 F7 :4B
70F0 19 35 10 CC 00 08 17 F7 11 17 F7 50 17 00 84 10 :5A
---------------------------------------------------------
Sum: 74 D7 F4 9F 42 D1 A8 7E C4 68 14 C0 51 1A 63 30 :15

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7100 26 00 BE 8E 04 3D A6 80 81 01 10 26 00 B3 E6 81 :AB
7110 10 AE 81 0D 35 26 02 EE 84 30 02 34 04 1F 30 34 :08
7120 06 17 F7 E1 86 2D 17 F7 0D 35 06 34 20 E3 E1 83 :99
7130 00 01 17 F7 D0 35 04 C0 05 A6 80 0D 3E 26 08 A1 :1D
7140 C0 10 26 00 7C 20 02 A7 C0 31 3F 27 0E 5A 26 E9 :09
7150 8D 31 26 6D 8E 04 3E E6 80 20 DE 63 80 26 62 30 :20
7160 02 86 2C 17 F6 D0 EC 84 17 F7 9A 30 8D FE CF CC :FF
7170 00 0E 17 F6 95 4F 97 52 86 01 97 50 16 F6 93 86 :7B
7180 FF 20 F3 86 03 97 50 8E 04 3D 8D 2C 81 01 26 FA :AC
7190 8D 26 81 3C 26 F4 8D 20 97 37 A7 80 8D 1A A7 80 :FA
71A0 27 15 1F 89 9B 37 97 37 8D 0E A7 80 9B 37 97 37 :E6
71B0 5A 26 F5 8D 03 91 37 39 34 10 17 F6 55 96 52 35 :C9
71C0 90 30 8D FE 87 CC 00 12 16 FF A7 30 8D FE 8F CC :82
71D0 00 0F 16 FF 9D 33 8D F5 69 30 8D F5 D2 31 8D FC :1D
71E0 07 86 04 97 36 8D 19 96 33 A7 C0 0A 36 26 F6 86 :16
71F0 05 97 36 8D 0B DC 32 ED C1 0A 36 26 F6 16 F9 62 :F3
---------------------------------------------------------
Sum: 34 78 41 E6 50 C3 09 30 C3 C7 02 1C 1C A2 AA DA :09

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7200 EC 81 10 A3 A1 26 10 A6 80 81 2D 26 0A 17 F7 32 :3B
7210 A6 80 81 20 26 01 39 32 62 16 FA FE 8E 04 3D 9F :37
7220 3E 30 8D F5 8B 17 F6 F8 C6 08 0F 36 A6 80 81 2C :66
7230 27 0B 17 00 BA 5A 26 F4 17 F6 F8 20 08 86 20 17 :61
7240 00 AD 5A 26 F8 17 F6 FA DC 32 1F 03 DD 3A 17 F6 :80
```

```
7250 E2 17 F6 EE DC 32 93 3A C3 00 01 1F 02 17 F6 D3 :7D
7260 17 F6 DF 86 FF 97 52 86 01 97 50 17 F5 A4 86 0A :08
7270 8E 00 00 30 1F 26 FC 4A 26 F6 0F 35 17 00 55 86 :9B
7280 02 8D 6C C6 0B 4F 8D 67 5A 26 FA 8D 76 5F 86 FF :70
7290 8D 4C 5A 26 FB 0C 35 4F 8D 55 1F 20 8D 4D DC 3A :F5
72A0 8D 49 A6 C0 8D 49 31 3F 26 F8 86 FF 8D 41 CC 00 :BF
72B0 00 8D 38 DC 32 8D 34 8E 04 3D 9C 3E 27 02 8D 43 :36
72C0 86 01 8D 1A 86 3C 8D 16 86 FF 8D 12 4F 8D 0F 8D :2F
72D0 03 16 FE A1 C6 0E 86 FF 8D 04 5A 26 FB 39 34 12 :9C
72E0 97 52 86 02 97 50 17 F5 29 35 92 8D 02 1F 98 34 :CE
72F0 10 9E 3E A7 80 9F 3E 0C 36 96 36 81 FF 26 02 8D :33
---------------------------------------------------------
Sum: CA AC 57 6E 26 08 CB 61 08 D2 97 18 33 10 55 49 :FF
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7300 02 35 90 34 10 8E 04 3D 9F 3E 86 01 8D D0 86 3C :5D
7310 8D CC 96 35 8D C8 97 37 96 36 0F 36 8D C0 1F 89 :4D
7320 9B 37 97 37 A6 80 8D B6 9B 37 97 37 5A 26 F5 96 :B4
7330 37 8D AB 17 FF 9E 35 90 17 F5 11 17 F6 00 DE 32 :22
7340 17 F5 F0 17 F5 FC DC 32 DD 3C 17 F5 E6 17 F5 F2 :1B
7350 10 9E 32 17 F5 CA A6 80 27 13 81 2C 10 26 F5 C9 :B7
7360 17 F5 BD A6 84 81 50 10 26 F5 BE 03 31 A6 C4 A1 :EC
7370 A4 27 24 1F 30 17 F5 8D 86 2D 17 F4 B9 A6 C4 17 :CF
7380 F5 87 17 F4 DD A6 A4 17 F5 7F 86 2D 17 F4 A7 1F :BD
7390 20 17 F5 71 17 F4 B5 17 F5 3E 81 03 26 01 39 31 :BC
73A0 21 33 41 11 93 3C 25 C5 39 16 F4 A0 16 F4 DF 16 :41
73B0 F5 90 16 F5 6B 16 F5 71 16 F5 1D 16 F4 78 16 F5 :2C
73C0 48 16 F5 41 16 F4 9B 16 F5 69 16 F4 87 16 F7 92 :DD
73D0 16 F8 F1 17 07 13 0F 31 8D CF 30 8D F3 7C 9F 38 :CF
73E0 0C 31 30 8D F3 CA 8D C7 DC 32 DD 3A 8D D9 8D BF :E2
73F0 DC 32 DD 3C 8D BC A6 80 27 0E 81 2C 26 B7 A6 84 :7F
---------------------------------------------------------
Sum: B4 46 C1 36 6F 4B 74 FB 55 51 66 6A BE C2 88 68 :00
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7400 81 50 26 B1 86 FF 97 31 DE 3A 8D 0E 8D AA 81 03 :63
7410 27 07 11 93 3C 25 F3 27 F1 39 34 36 17 00 92 17 :A1
7420 FF A8 30 8D F2 DA EC 81 17 FF 96 17 FF 96 0D 29 :2B
7430 27 07 EC 81 17 FF 8A 20 0B A6 81 17 FF 80 17 FF :39
7440 83 17 FF 80 17 FF 7D C6 07 96 28 27 0E 34 02 A6 :48
7450 80 17 FF 6A 35 02 5A 5A 4A 26 F0 5D 27 06 17 FF :EB
7460 63 5A 26 FA A6 80 81 20 27 04 5C 17 FF 4D 31 8D :4C
7470 F2 9B 34 20 AC E1 26 EC C1 04 27 14 A6 80 81 20 :47
7480 27 04 17 FF 36 5C C1 04 27 06 17 FF 37 5C 20 F6 :84
7490 17 FF 31 17 FF 2E A6 80 81 20 27 03 17 FF 1C 31 :DF
74A0 8D F2 84 34 20 AC E1 26 ED 17 FE FD 17 FF 21 35 :75
74B0 B6 30 8D F2 4B 86 20 C6 27 A7 80 5A 26 FB 86 01 :6C
74C0 97 28 17 00 0F 96 27 4A 48 30 8D 04 4B EC 86 1F :D1
74D0 51 AD 8B 39 31 8D F2 28 EF A1 0F 29 E6 C0 E7 A0 :8F
74E0 C1 10 26 0A 0D 29 10 26 01 0C 0C 29 20 EE C1 11 :8F
74F0 26 0C 0D 29 10 26 00 FE 0C 29 20 EE 0C 29 30 8D :D1
---------------------------------------------------------
Sum: 76 3F D9 FE 66 8D 0F 2B 2A C6 F7 BE 64 DF 43 4E :32
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7500 04 E7 3A E6 84 C1 B8 10 27 00 A6 0D 29 10 27 00 :52
7510 EE C1 54 26 1A D6 03 54 10 24 00 95 30 8D 04 C9 :C3
7520 C6 81 3A 0A 29 26 03 86 44 8C 86 55 5F 20 68 C1 :B6
7530 57 26 0D 0A 29 26 03 86 32 8C 86 33 C6 03 20 57 :23
7540 96 03 84 F0 81 20 26 19 0A 29 10 26 00 A6 96 03 :95
7550 84 0F 10 27 00 9E CC 4C 42 DD 07 C6 0A 86 20 20 :3C
7560 36 96 03 81 80 25 4A 96 03 84 0F 81 0E 24 16 81 :B5
7570 0C 26 3E E6 84 C1 18 26 38 0A 29 26 03 86 59 8C :D8
7580 86 53 5F 20 12 96 03 84 F0 81 C0 24 03 86 59 8C :4A
7590 86 53 5F 0A 29 26 5D 0C 29 97 0D D7 27 4F E6 84 :7E
75A0 30 8D 03 8A 3A 31 8D F1 60 A6 80 2B 06 A7 A0 20 :51
75B0 F8 20 43 84 7F A7 A0 0D 27 26 29 0D 29 26 04 96 :1E
75C0 02 20 02 96 03 84 F0 26 04 86 06 20 15 44 44 44 :E8
75D0 44 84 03 26 10 86 04 D6 0D C1 41 27 05 C1 42 27 :C6
75E0 01 4C 97 27 39 81 01 27 E0 81 02 26 03 86 08 8C :93
75F0 86 07 20 EE 0C 29 C6 0B D7 27 4F C6 B8 20 A1 D6 :03
---------------------------------------------------------
Sum: 6C 67 6A A7 C1 CF 5D 4D 9C A3 0F 23 C7 E3 EA A4 :C7
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7600 02 C1 80 25 48 C1 C0 25 03 86 42 8C 86 41 C4 0F :47
7610 C1 03 26 04 86 44 20 31 C1 0C 25 2D C1 0D 26 12 :2E
7620 E6 84 C1 64 27 EE C1 91 27 02 5F 8C C6 09 86 20 :7F
7630 20 18 D6 02 C1 C0 24 04 86 58 20 0D D6 02 C4 0F :6F
7640 C1 0E 24 03 86 44 8C 86 55 5F 16 FF 4C C4 F0 C1 :5C
7650 60 24 0F C1 40 25 0E 26 03 86 41 8C 86 42 C6 03 :D4
7660 20 E8 5F 20 C9 C1 10 24 04 C6 06 20 C1 C1 20 26 :FD
7670 06 86 42 97 08 20 B5 C1 30 26 21 D6 02 C4 0F C1 :E6
7680 04 24 04 C6 08 20 A7 C1 08 24 04 C6 01 20 9F C1 :F9
7690 0C 26 03 C6 04 8C C6 03 86 20 20 AE D6 02 C4 0F :73
76A0 C1 0E 25 04 C6 02 20 86 C1 0D 27 EA C1 0A 25 06 :3B
76B0 C6 04 86 43 20 94 C1 09 27 DC C1 06 25 D8 C6 0A :A8
76C0 20 D6 E6 C0 D7 04 0F 2C 34 04 54 54 54 54 31 8D :F8
76D0 F0 3C 31 21 8D 19 C6 2C E7 A0 35 04 30 01 C4 0F :DA
76E0 8D 0D 04 2C 25 01 39 CC 2A 2A DD 0F DD 12 39 34 :91
76F0 10 30 8D 01 FB C1 06 25 02 0C 2C C1 0C 25 02 C6 :A9
---------------------------------------------------------
Sum: 54 AB 6B EB C3 1E 86 18 BA C4 02 5F A2 74 97 71 :D1
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7700 06 58 3A EC 81 ED A1 35 90 A6 C0 97 04 31 8D 01 :18
7710 CF 30 8D EF F9 C6 08 D7 2D 44 34 02 24 21 EC A1 :92
7720 34 02 96 2D 81 02 26 0E 96 02 84 0F 81 06 25 06 :8D
7730 35 02 86 53 20 02 35 02 ED 81 86 2C A7 80 8C 31 :6D
7740 22 35 02 0A 2D 26 D2 86 20 A7 82 39 A6 C0 97 04 :91
7750 8D 46 DD 13 20 18 8D F4 DD 11 A6 C0 97 05 0C 28 :A0
7760 20 EE 86 23 97 0F 20 E4 86 23 97 0F 20 E8 34 02 :EE
7770 86 24 97 10 35 82 E6 C0 D7 04 1D DF 2A D3 2A 34 :E0
7780 04 8D 15 DD 11 35 02 8D 0F DD 13 20 E1 EC C1 DD :E2
7790 04 0C 28 20 E6 0A 28 39 8D 09 1E 89 8D 10 1E 89 :2A
77A0 8D 0C 39 1F 89 84 0F 1E 89 44 44 44 44 39 81 0A :88
77B0 25 02 8B 07 8B 30 39 86 2D 97 0F 39 A6 C4 84 60 :8D
77C0 27 11 81 20 27 0A 81 40 27 03 86 53 8C 86 55 8C :C1
77D0 86 59 8C 86 58 97 18 86 2C 97 15 A6 C0 97 04 2B :82
77E0 14 84 1F 1F 89 58 58 58 2A 08 8D CB 50 54 54 54 :3D
77F0 1F 98 16 FF 5B 84 0F 31 8D 01 0D E6 A6 4F 1F 52 :D2
---------------------------------------------------------
Sum: 2D 46 22 92 A2 F6 DB F3 F6 B0 93 8B 71 11 DB 68 :16
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7800 6E AB 86 2B 97 19 17 00 96 10 26 00 8D 39 CC 2B :1A
7810 2B DD 19 20 0B 86 2D 97 16 20 EB CC 2D 2D DD 16 :D0
7820 20 7D 86 42 97 12 20 77 86 41 20 F8 A6 C0 97 05 :86
7830 2A 05 C6 2D D7 0F 40 17 FF 16 0C 28 20 61 D7 05 :05
7840 EC C1 DD 05 2A 0D 50 26 01 4A 43 34 02 86 2D 97 :4A
7850 0F 35 02 34 04 17 FF 40 DD 11 35 02 17 FF 39 DD :25
7860 13 17 FF 0A 0C 28 20 D2 86 44 20 B8 8D 17 8D CA :F6
7870 E6 C0 D7 05 1D 16 FF 03 8D 0B 8D BE 0C 28 EC C1 :7B
7880 DD 05 17 FE F6 CC 50 43 DD 18 39 EC C1 DD 05 8D :96
7890 C2 27 07 86 20 97 18 97 15 39 CC 2A 2A 20 E9 34 :87
78A0 02 96 04 85 10 26 02 35 82 86 5B 97 0E 86 5D 97 :10
78B0 1B 35 82 86 27 97 0E 97 10 D6 02 8D 09 D7 0F 0A :29
78C0 28 0D 29 26 0C 39 C1 20 2B 04 C1 60 2B 02 C6 2E :1B
78D0 39 97 12 97 14 86 2C 97 11 D6 03 8D E9 D7 13 39 :59
78E0 43 43 41 20 42 20 44 50 58 20 59 20 55 20 50 43 :D6
78F0 44 20 58 20 59 20 55 20 53 20 50 43 2A 2A 2A 2A :78
---------------------------------------------------------
Sum: 7B D5 18 8E 6F 41 10 2D 8D F8 31 22 C7 C8 A3 80 :6D
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7900 41 20 42 20 43 43 44 50 02 0E 15 1B 9F 22 28 9A :A0
7910 2C 40 9A 68 6C 78 9A 8B 02 38 01 F1 02 C4 02 91 :FC
7920 02 97 02 7B 02 85 02 EB 02 A5 02 BC 03 E2 41 42 :57
7930 D8 41 44 C3 41 44 C4 41 4E C4 41 53 CC 41 53 D2 :82
7940 42 49 D4 43 4C D2 43 4D D0 43 4F CD 44 41 C1 44 :09
7950 45 C3 45 4F D2 45 58 C7 49 4E C3 4A 4D D0 4A 53 :30
7960 D2 4C 53 CC 4C 53 D2 4D 55 CC 4E 45 C7 4E 4F D0 :E3
7970 52 4F CC 52 4F D2 52 54 C9 52 54 D3 53 42 C3 53 :73
7980 45 D8 53 55 C2 53 57 C9 54 46 D2 54 53 D4 4C C4 :F1
7990 4F D2 53 D4 43 C3 43 D3 45 D1 47 C5 47 D4 48 C9 :B2
79A0 4C C5 4C D3 4C D4 4D C9 4E C5 50 CC 52 C1 52 CE :C8
79B0 53 D2 56 C3 56 D3 43 57 41 C9 53 59 4E C3 4C 42 :56
79C0 53 D2 4C 42 52 C1 4C 45 41 D8 4C 45 41 D9 4C 45 :AC
79D0 41 D3 4C 45 41 D5 50 53 48 D3 50 53 48 D5 50 55 :DE
79E0 4C D3 50 55 4C D5 46 43 C3 3C B8 B8 1B 36 B8 45 :2B
79F0 0F 0C 42 21 B8 2A 5D 2D 15 B8 B8 3F 8C B8 B8 94 :3E
---------------------------------------------------------
Sum: 14 A4 CC 32 E9 12 CC 80 14 A2 D5 17 85 72 19 09 :B8
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7A00 90 B8 1E 62 B8 09 51 27 5A 7E 80 70 74 66 68 7A :85
7A10 6A 84 86 7C 78 6C 76 6E 72 98 9C A0 A4 A8 B0 AC :A6
7A20 B4 B8 4B 00 48 88 39 B8 57 3C B8 B8 1B 36 B8 45 :C9
7A30 0F 0C 42 21 B8 2A 5D B8 15 3C B8 B8 1B 36 B8 45 :84
7A40 0F 0C 42 21 B8 2A 5D B8 15 3C B8 B8 1B 36 B8 45 :84
7A50 0F 0C 42 21 B8 2A 5D 2D 15 3C B8 B8 1B 36 B8 45 :F9
7A60 0F 0C 42 21 B8 2A 5D 2D 15 54 18 4E 54 09 12 60 :88
7A70 B8 24 03 62 06 18 91 60 B8 54 18 4E 54 09 12 60 :91
7A80 64 24 03 62 06 18 30 60 64 54 18 4E 54 09 12 60 :88
7A90 64 24 03 62 06 18 30 60 64 54 18 4E 54 09 12 60 :88
7AA0 64 24 03 62 06 18 30 60 64 54 18 4E 06 09 12 60 :3A
7AB0 B8 24 03 62 06 60 B8 60 B8 54 18 4E 06 09 12 60 :B2
7AC0 64 24 03 62 06 60 64 60 64 54 18 4E 06 09 12 60 :B6
7AD0 64 24 03 62 09 60 64 60 64 54 18 4E 06 09 12 60 :B9
7AE0 64 24 03 62 09 60 64 60 64 39 4A 49 31 45 54 46 :5A
7AF0 20 49 20 4C 4F 56 45 20 48 49 52 4F 4B 4F 2E 20 :F9
---------------------------------------------------------
Sum: D2 8D 2F BE DD DB BE 37 87 28 58 F8 68 C2 0A A0 :CC
```

FM-8

6809 SUPER TRACER V.2.9

■Mr.かれきさん

FM-8でTiny Forthを開発中，JMP，LEAなどの命令とサブルーチン管理，パラメータ・チェックに苦しみ，EAとスタックのトップ1ワードを表示するデバッガを開発したのでご紹介します．

主な特色＋ただの特色

I　EA注1，(EA) 両スタックのTOP1WORDおよびマシン・コードの表示．
　注1：EA…Enable Addressの略

II　コマンド入力時の16進4桁入力時に各レジスタ値およびEA，字数を1キーで入力可能(〈形式1〉のところ)

III　サブルーチンを登録でき，登録されたルーチンがcallされると“SUBUTINE”と表示され，アドレスが解りサブルーチンの使用が一目瞭然です．

IV　プログラム領域上限の設定可．

V　PTMを使わないので，ハードウェアに依存しない．

VI　コマンド追加が容易．

VII　ページ・リロケータブル（ワーク・エリアも含めて）なので，どこへでも移せる．

VIII　コンパクト（ワークを含め3Kバイト）

IX　ブレーク・ポイントとトレースとの両刀使いができる．

X　他機種への移植が容易．

使い方

WIDTH80 (20でも，25でもご自由に) としてください．40字モードでは，表示とメモリ・ダンプでちぐはぐになってしまいます．

先に述べたとおり，トレースとブレークと2つの使い方があるので適宜使い分けてください．以下に詳細を述べます．

共通命令

❶プリンタ・サポート：o,c

oはもちろんオープンのつもりですが，実際にはイニシャライズするでもなく，単にOUCHAをコールしたとき，直接BIOS内ののルーチンをcallするだけです．

そのため，接続していないときにcallすると止まってしまいます．なお，cはクローズです．

❷モニタJMP：E

SWIのベクトルを書き変えて，モニタへ戻ります．

表1．命令一覧表

BREAK POINT 関係

B		ON	BK SET	BKを OFF
		OFF	BK CLR	
		4HEX	登録	
		C/X	以下抹消	
		RETURN	戻り	
		その他	NEXT	
g			即実行	BKを ON
G	★	Y	実行	
		その他	とりやめ	
共通				
O			PRINTER OPen	
C			PRINTER Close	
@	4HEX			
E			EXIT	

共通

R			レジスタ表示	
r		2H		
		4HEX		
		RETURN	終了	
		␣	NEXT	
		その他	再入	
D	★		1ページ	STOP で 中断
d	★	␣★	★から★まで ずっと	
M	★	↑	DEC	
		␣	INC	
		2H		
		K		
		RETURN	終了	
		その他	再入	

トレース関係

●			シングルSTEP
T	2H		
	他	BEEP	
S	S	4HEX	登録
	C	4HEX	抹消
	D		ダンプ
m			マクロ.OM/OFF
X	4HEX		ROMSTT

★：アキュムレータ　　@　字数
　X　Xレジスタ　　I　EA
　Y　Yレジスタ　　4HEX
　U　Uポインタ
　S　Sポインタ
　P　Pカウンタ

❸レジスタ表示：R

ここが最大の特色で，他はすべておまけといってよいほど主要な部分を成しています．

CODEとはマシン語のことで，MPUの最大語長の5バイト表示します．

EFHINZVCとはCCRのことで，な，なんとバイナリで表示します．クリティカルな点やきめの細かなプログラム作りには強い身方となるでしょう．

以下おきまりのレジスタ群が続きましてSTOP,UTOPの各々1ワード，そしてEA,(EA)が出てきます．STOPの表示はサブルーチン内でのローカル変数の管理に，UTOPはパラメータに，EAはJMP，インダイレクト，インデックス命令などのバグ取りに役立つでしょう．

❹レジスタ変更：r

変更なしは SPACE キー， RETURN で終了です．

❺メモリ・チェンジ：M〈形式1〉

〈形式1〉とは，1ワード入力のことでRでDレジスタ，X，Y，S，U，Pはそれぞれのレジスタ，@は定数，IはEA，¥は直前に使ったアドレス，16進級ならそのもの，となっています．

SPACE で次に進み， ↑ で戻り， RETURN で終了， K でASCII入力となっています．

❻ 字数変更：@

そのものです．

ブレーク・ポイント

❶実行：G〈形式1〉，g

Gではアドレスが表示されるので，内容を確かめた上で Y を押してください．

gは現在のPCからで，すぐに実行します．

②ブレーク・ポイント：B

登録済のBP（ブレーク・ポイント）を表示し，次のコマンドを受け付けます．

C/X：以下をキャンセルします．

ON：BPをSETします．

OF：BPをクリアします（Bが呼ばれた時点ですでにクリアするので無意味）．

16進キー：BPの変更．

RETURN キーでコマンド終了．

トレース

❶シングル・ステップ：

上限領域に入ると，BEEPして戻ります．またサブルチンをcallするとき登録されていれば，表示してトレースを一時中断します．

❷トレース：T〈2HEX〉

2桁を入力し，その回数だけトレースします．なお，00のときは256を意味し， STOP キーでアボードします．

❸領域設定：X〈4HEX〉

❹サブルーチン登録：S(D,S,C)

SD：登録中のものをダンプします．

SS：登録

SC：抹消

雑記

JMP，XなどはLEAPC，Xのように，LEAPCと変えて考えればJMP命令時のインデックスの怪とか，インダイレクトの亡霊，INDIRECTプラスオート・インクリメントの錯乱からの脱出が可能と思われます．

SWIベクトル($FFFA−B)を$AFA7と変更すれば，いとも簡単にブレーク処理が楽しめます(一度やってみてくださいな！なぜF-BASICの文法書に書かないのか不思議なくらいです)．

表2

パッチ・エリア	アドレス	元のデータ	変更後
プリンタで，CR/LFのとき，CRで改行されないものは，	5226	27	21
INCHA0 1文字→AccA	5251	D336	
OUCHA0 AccA→表示	5233	D352	
モニタ・ジャンプ	595E	AFA5	
	5468 5969	AFA7	
SWI ベクトル	513F 5961	FFFA	
INCHA ルーチン	524B →5259		
OUCHA ルーチン	521E →5236		
OUTPTR AccA→プリント	522D	F935	

フラグは
<$1B…プリンタ…0＝なし
<$1C…エコー……0＝あり

コマンド追加 51ABから51B7まで

5
51AB　コマンド名　一文字　　アドレス−5178
AE
B1
B4
B7　0 0→おわりに必ず入れる．

おわりに

このプログラムを活用され，皆さんのマシン語プログラムの開発への手助けとなれば幸いです．

実行例

```
>
>R
ADDR CODE       EFHINZVC A B  DP  X    Y    U    PC   S    STOP UTOP EA   (EA)
6000 1605270000 00010100 005A 00 CAA5 C830 0340 6000 70A5 C9C6 5452 652A 17FF
```

実行例

```
>T20

ADDR CODE       EFHINZVC A B  DP  X    Y    U    PC   S    STOP UTOP EA   (EA)
652A 17FF5DA68D 10010100 005A 00 CAA5 C830 0340 652A 70A5 C9C6 5452 648A 308D
648A 308D03B016 10010100 005A 00 CAA5 C830 0340 648A 70A3 652D 5452 683E 4449
648E 16FEC5E680 10010000 005A 00 683E C830 0340 648E 70A3 652D 5452 6356 A680
6356 A680810427 10010000 005A 00 683E C830 0340 6356 70A3 652D 5452 683E 4449
6358 810427048D 10010000 445A 00 683F C830 0340 6358 70A3 652D 5452
635A 27048DBE20 10010000 445A 00 683F C830 0340 635A 70A3 652D 5452 635C 8DBE
635C 8DBE20F639 10010000 445A 00 683F C830 0340 635C 70A3 652D 5452 631C 16FD
631C 16FD198D12 10010000 445A 00 683F C830 0340 631C 70A1 635E 5452 6038 347D
6038 347DBD685F 10010000 445A 00 683F C830 0340 6038 70A1 635E 5452
603A BD685FBE68 10010000 445A 00 683F C830 0340 603A 7098 905A 5452 685F 3408
685F 3408C6001F 10010000 445A 00 683F C830 0340 685F 7096 603D 5452
6861 C6001F9BC6 10010000 445A 00 683F C830 0340 6861 7095 0060 5452
6863 1F9BC614BD 10010100 4400 00 683F C830 0340 6863 7095 0060 5452
6865 C614BD6891 10010100 4400 00 683F C830 0340 6865 7095 0060 5452
6867 BD68913508 10010000 4414 00 683F C830 0340 6867 7095 0060 5452 6891 3416
6891 34168E68A5 10010000 4414 00 683F C830 0340 6891 7093 686A 5452
6893 8E68A5A706 10010000 4414 00 683F C830 0340 6893 708F 4414 5452
6896 A706E784CC 10010000 4414 00 68A5 C830 0340 6896 708F 4414 5452 68AB 4400
6898 E784CC0001 10010000 4414 00 68A5 C830 0340 6898 708F 4414 5452 68A5 1400
689A CC0001ED04 10010000 4414 00 68A5 C830 0340 689A 708F 4414 5452

ADDR CODE       EFHINZVC A B  DP  X    Y    U    PC   S    STOP UTOP EA   (EA)
689D ED04AD9FFB 10010000 0001 00 68A5 C830 0340 689D 708F 4414 5452 68A9 0001
689F AD9FFBFA35 10010000 0001 00 68A5 C830 0340 689F 708F 4414 5452 F2D8 347F
SUBRUTIN F2D8

68A3 3596140068 10010000 0001 00 68A5 C830 0340 68A3 708F 4414 5452 686A 3508
686A 3508393408 10010000 4414 00 683F C830 0340 686A 7095 0060 5452
686C 393408C600 10010000 4414 00 683F C830 0340 686C 7096 603D 5452 603D BE68
603D BE686D35FD 10010000 4414 00 683F C830 0340 603D 7098 905A 5452 686D 3408
6040 35FD347DBD 10010000 4414 00 3408 C830 0340 6040 7098 905A 5452 635E 20F6
635E 20F639862D 10010000 445A 00 683F C830 0340 635E 70A3 652D 5452 6356 A680
6356 A680810427 10010000 445A 00 683F C830 0340 6356 70A3 652D 5452 683F 4953
6358 810427048D 10010000 495A 00 6840 C830 0340 6358 70A3 652D 5452
635A 27048DBE20 10010000 495A 00 6840 C830 0340 635A 70A3 652D 5452 635C 8DBE
635C 8DBE20F639 10010000 495A 00 6840 C830 0340 635C 70A3 652D 5452 631C 16FD
```

6809 SUPER TRACER マシン語ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5100 20 09 20 07 1F 50 80 01 1F 8B 39 34 FF 8D F5 10 :E8
5110 DF 0A 5F DD 00 C6 86 DD 02 C3 00 10 DD 04 C3 00 :C7
5120 10 DD 06 C3 00 10 DD 08 31 8D 02 7C 9E 00 30 88 :3D
5130 18 C6 20 A6 A0 A7 80 5A 26 F9 30 8D 03 1C BF FF :7E
5140 FA AE A4 9F 0C 30 8D 01 62 17 00 9C 10 9E 0A 17 :99
5150 00 8C 86 3E 17 00 C7 17 00 F1 30 8C 19 30 02 E6 :23
5160 80 26 07 86 07 17 00 B6 20 E5 34 04 A1 E0 26 ED :D8
5170 EC 84 30 8D 00 02 6E 8B 2E 06 49 54 06 96 52 06 :ED
5180 A2 72 07 0C 4D 07 94 45 07 E5 40 07 F3 44 08 22 :E8
5190 64 07 FE 47 08 F5 67 08 F3 6F 09 13 63 09 18 42 :60
51A0 09 C5 53 09 28 58 09 21 6D 09 1C 00 AE 87 58 AE :A1
51B0 87 73 AE 87 6D AE 87 00 86 08 8D 62 86 05 8D 5E :C4
51C0 9E 06 17 00 86 81 08 27 F3 9C 08 24 EB A7 80 81 :3F
51D0 03 27 05 81 0D 26 EB 39 10 DE 0A 7E AF A7 34 16 :1D
51E0 30 8D 01 3D 8D 02 35 96 A6 80 81 04 27 05 17 00 :43
51F0 2D 20 F5 39 8D 0C 8D 0A 8D 08 8D 06 8D 04 86 20 :0A
---------------------------------------------------------
Sum: 21 25 1E 17 80 CD 65 07 4B 2E 2A F5 25 21 81 AE :41


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5200 20 1C A6 80 20 04 8D 02 1F 98 34 02 44 44 44 44 :12
5210 8D 04 35 02 84 0F 8B 30 81 39 23 02 8B 07 34 0E :C9
5220 0D 1B 27 0B 81 0D 27 07 C6 FD 1E B9 BD F9 35 5F :FA
5230 1E 9B BD D3 52 35 8E A6 80 8D 00 8D 00 8D 00 48 :73
5240 34 02 86 30 24 01 4C 8D D5 35 82 34 08 4F 1F 8B :AB
5250 BD D3 36 35 08 0D 1C 27 C5 39 8D EF 8D 34 25 18 :CB
5260 48 48 48 48 34 02 8D E3 20 1D 8D EE 25 0A 1F 89 :55
5270 8D E8 25 04 1E 89 1C FE 39 A6 80 8D 15 25 10 48 :DD
5280 48 48 48 34 02 A6 80 8D 09 25 05 AB E0 1C FE 39 :D2
5290 35 82 80 30 25 12 81 0A 25 0B 80 07 25 0A 81 10 :A0
52A0 24 06 81 0A 39 1C FE 39 1A 01 39 36 38 30 39 20 :8C
52B0 53 55 50 45 52 20 54 52 41 43 45 52 20 20 56 32 :38
52C0 2E 39 20 28 43 29 20 4D 72 2E B6 DA B7 20 BB DD :27
52D0 0D 0A 04 41 44 44 52 20 43 4F 44 45 20 20 20 20 :F1
52E0 20 20 20 45 46 48 49 4E 5A 56 43 20 41 20 42 20 :A0
52F0 20 44 50 20 20 58 20 20 20 20 59 20 20 20 20 55 :FA
---------------------------------------------------------
Sum: 0D A7 15 92 94 EF 0C 71 91 F3 2A 81 F0 79 6B 7A :D8


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5300 20 20 20 20 50 43 20 20 20 53 20 20 20 20 53 54 :ED
5310 4F 50 20 55 54 4F 50 20 20 45 41 20 20 28 45 41 :BB
5320 29 0D 0A 00 00 00 04 0D 0A 20 20 43 43 3D 20 20 :9E
5330 04 20 20 44 3D 20 04 20 20 44 50 3D 20 04 0D 0A :35
5340 20 20 58 3D 20 04 20 20 59 3D 20 04 20 20 55 3D :C5
5350 20 04 20 20 50 43 3D 20 04 0D 0A 20 20 53 54 4F :A5
5360 50 20 20 3D 20 04 20 20 55 54 4F 50 20 20 3D 20 :16
5370 04 0D 0A 20 20 43 4F 44 45 30 31 3D 20 04 20 20 :78
5380 43 4F 44 45 32 33 3D 20 04 20 20 43 4F 44 45 34 :70
5390 3D 20 04 0D 0A 4D 41 43 52 4F 20 04 0D 0A 53 55 :CD
53A0 42 52 55 54 49 4E 20 04 00 00 00 00 00 00 00 00 :F8
53B0 20 03 1C FE 39 1A 01 39 00 00 00 00 00 00 D3 36 :D3
53C0 D3 52 F9 35 F2 D8 00 00 80 00 81 01 81 01 01 01 :A3
53D0 01 00 02 03 00 01 02 03 00 00 02 00 00 02 02 00 :12
53E0 02 02 02 02 02 00 02 02 82 02 01 01 01 01 00 00 :96
53F0 83 03 00 01 02 00 02 01 82 82 12 12 12 12 12 12 :FC
---------------------------------------------------------
Sum: 6B 09 C2 52 45 01 E9 B7 3B BD 51 CC 13 84 4B 5D :C2


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5400 12 12 12 12 12 12 12 12 12 12 40 40 40 40 02 82 :38
5410 02 82 00 81 01 81 02 01 00 81 01 00 00 01 01 00 :0E
5420 01 01 01 01 01 00 01 01 00 01 01 00 00 01 01 00 :0B
5430 01 01 01 01 01 00 01 01 00 01 40 00 00 40 40 00 :C8
5440 40 40 40 40 40 00 40 40 C0 40 20 00 00 20 20 00 :20
5450 20 20 20 20 20 00 20 20 A0 20 17 FC A7 10 DF 0A :53
5460 0D 18 27 0D 10 DE 0A 7E AF A7 00 00 00 00 00 00 :25
5470 00 10 9E 0A 17 FD 67 0D 1E 26 0E 17 FD 60 30 8D :C3
5480 FE 51 17 FD 63 86 14 97 1E 0A 1E AE 2A 30 1F AF :13
5490 2A DE 02 0F 19 AC C1 26 0B A6 4E 81 3F 27 05 A7 :57
54A0 B8 0A 20 14 AC C1 27 07 11 93 04 25 F7 20 B5 EC :16
54B0 4E A7 B8 0A E7 4E 03 19 AE 2A 1F 10 17 FD 47 17 :81
54C0 FD 3C 17 FD 2F 1F 21 17 FD 6D 17 FD 31 17 FD 2A :C0
54D0 17 FD 29 17 FD 24 17 FD 21 17 FD 1E 17 FD 1B 1F :2A
54E0 10 17 FD 22 17 FD 17 17 FD 14 17 FD 0D AE 28 17 :A7
54F0 FD 08 0F 14 0F 15 0F 16 0F 17 0F 1F AE 2A EC 04 :8D
---------------------------------------------------------
Sum: D2 56 76 80 FD 04 44 1E 51 DE 90 EE 5E 72 BF D6 :93


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5500 DD 2C EC 02 DD 2A EC 84 DD 28 10 2A 01 38 81 8D :F4
5510 26 0D 03 14 86 02 97 1F 86 20 97 28 16 01 57 81 :DC
5520 CD 10 27 02 BD 84 4F 81 0D 26 02 03 14 86 30 94 :AD
5530 28 26 29 C6 02 86 0F 94 28 81 03 27 12 81 07 10 :E5
5540 27 02 9F 81 0C 27 08 81 0E 10 22 02 95 26 02 C6 :CA
```

```
5550 03 DB 1F D7 1F 0F 0E 0F 0F 16 02 3E 80 10 26 0D :47
5560 A6 23 DD 0E C6 02 DB 1F D7 1F 16 02 2D 80 10 27 :68
5570 0D DC 29 DD 0E C6 03 DB 1F D7 1F 16 02 1C 0C 1F :15
5580 5D 2B 15 34 04 8D 26 35 04 58 58 58 57 57 57 30 :FE
5590 85 9F 0E 0C 1F 16 02 02 C5 10 27 02 03 15 C1 9F :ED
55A0 26 24 9E 2A C6 03 DB 1F D7 1F 16 00 8E C4 60 26 :B9
55B0 02 AE 24 C1 20 26 02 AE 26 C1 40 26 02 AE 28 C1 :71
55C0 60 26 02 30 2C 39 34 04 8D E3 35 04 C4 0F C1 0C :9E
55D0 25 04 AE 2A 03 16 33 8D FD F0 A6 C5 10 27 02 02 :6D
55E0 2A 08 0D 15 10 26 01 FA 84 7F 9B 1F 97 1F 0D 16 :1B
55F0 26 38 C1 02 25 45 27 2E C1 04 25 28 27 3D C1 05 :1C
-----------------------------------------------------------
Sum: B4 51 66 BD 8E BA 69 FF 40 A9 75 64 FD 82 84 AA :47


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5600 27 1C C1 08 25 14 27 0E C1 09 27 04 EC 21 20 02 :9E
5610 DC 2A 30 8B 20 25 96 2A 20 06 A6 21 20 02 A6 22 :9D
5620 30 86 20 17 30 1F 30 1F 20 11 96 1F 30 86 C1 0C :F4
5630 26 05 D6 2A 1D 20 02 DC 2A 30 8B 0D 15 2A 02 AE :27
5640 84 9F 0E 16 01 54 33 8D FD 90 E6 C6 10 27 01 92 :5F
5650 2A 02 03 17 81 17 26 0E 03 14 DE 29 DF 2A C6 03 :02
5660 D7 1F 86 20 20 17 81 16 27 F0 34 04 C4 03 D7 1F :76
5670 35 04 C5 10 27 28 D6 29 1D DD 2A 96 28 97 20 AE :A3
5680 2A 96 1F 30 86 86 F0 97 17 AA A4 34 01 1F 8A 1F :04
5690 10 9D 20 24 02 D3 2A DD 0E 35 01 16 00 FC 85 F0 :98
56A0 26 07 0F 1F DC 28 16 FE B7 C5 20 10 26 FE C2 C5 :CA
56B0 40 27 05 DC 28 16 FE C6 DC 28 81 10 10 27 00 B7 :CD
56C0 81 11 10 27 00 C0 0F 0E 0F 0F 0D 17 10 27 00 CA :E9
56D0 81 39 26 07 AE 2C 9F 0E 16 00 BF 81 3B 26 10 6D :A2
56E0 2C 2B 04 AE 2D 20 03 AE A8 16 9F 0E 16 00 AB 81 :B4
56F0 3F 26 21 0F 17 0D 1A 10 27 00 9F 17 FA E0 30 8D :57
-----------------------------------------------------------
Sum: 20 91 F1 6B D9 D2 98 1F 1B B2 60 01 BE 2B 03 10 :99


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5700 FC 91 17 FA E3 AE 2A 30 01 17 FA FE 17 FA EF 0C :A5
5710 1F 16 00 86 81 35 27 04 81 37 26 29 0F 17 58 10 :31
5720 24 00 77 03 17 4F 8D 19 8D 15 48 8D 12 8D 10 C6 :96
5730 35 AE 28 D1 28 26 02 30 2C AE 86 9F 0E 20 5B 8D :71
5740 00 58 89 00 39 0F 17 34 04 C4 0F C1 05 35 04 10 :5A
5750 26 00 47 03 17 C4 F0 C1 05 10 22 00 85 30 8D 00 :75
5760 10 E6 85 2B 06 AE A5 9F 0E 20 2F 30 2C 9F 0E 20 :24
5770 29 01 04 06 08 80 0A C4 F0 C1 20 26 09 96 29 C6 :0F
5780 04 D7 1F 16 FE F7 DC 29 DD 28 9E 2B 9F 29 81 3F :60
5790 10 26 FD 7A 0F 0E 0F 0F 0C 1F 7D 03 13 27 05 0F :E1
57A0 1D 7F 03 13 DC 0E 27 0B 1F 01 17 FA 59 17 FA 4E :B7
57B0 17 FA 47 0D 19 10 26 F9 96 0D 1D 10 27 F9 90 0A :37
57C0 1D 0D 14 26 29 0D 17 26 21 AE 2A 96 1F 30 86 9F :DA
57D0 86 9C 0C 24 0D A6 84 97 96 86 3F A7 84 10 DE 0A :9E
57E0 35 FF 86 07 17 FA 37 16 F9 65 9E 0E 20 E1 9E 0E :D6
57F0 DE 00 33 C8 2E EC C1 27 F1 AC 5E 26 F8 30 8D FB :AC
-----------------------------------------------------------
Sum: D1 B2 4E 51 7E 15 61 0B 81 60 22 13 F2 09 19 BD :08


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5800 9B 17 F9 E4 DC 0E 17 F9 FD 17 F9 D2 20 BB 17 FA :54
5810 49 10 25 FF CD 4A 97 1D 20 A7 17 F9 C1 30 8D FA :97
5820 B2 17 F9 C4 16 FC 91 17 F9 BE 1E 13 17 F9 CD 17 :1C
5830 FA 19 81 0D 27 0E 81 20 27 07 17 FA 1F 25 09 A7 :AA
5840 1F 1E 31 39 1A 01 20 F9 86 07 17 F9 D1 86 09 17 :EF
5850 F9 CC 20 DB 17 F9 91 1E 13 17 F9 9E 17 F9 EC 81 :BD
5860 0D 27 11 81 20 27 0A 17 F9 F2 17 F9 FF 25 09 ED :43
5870 1E 1E 31 39 1A 01 20 F9 86 07 17 F9 A1 86 09 17 :BE
5880 F9 9C 20 D8 17 F9 57 30 8D FA 9C 1F 23 8D 98 25 :D3
5890 2F 8D C1 25 2B 8D 90 25 27 8D B9 25 23 8D B5 25 :2B
58A0 1F 8D B1 25 1B 8D AD 25 17 8D A9 25 13 EE 28 8D :24
58B0 A3 25 0D EE 2A 8D 9D 25 07 8D 99 25 03 17 FF 67 :0E
```

```
58C0 16 FF 57 17 F9 85 81 50 26 02 AE 2A 81 55 26 02 :D0
58D0 AE 28 81 59 26 02 AE 26 81 58 26 02 AE 24 81 52 :52
58E0 26 02 AE 21 81 40 26 02 9E 10 81 53 26 02 30 2C :E6
58F0 81 5C 26 02 9E 12 81 49 26 02 9E 0E 17 F9 93 25 :1B
-----------------------------------------------------------
Sum: 28 E6 76 25 16 FD A2 D4 92 A7 0D 7C 67 C6 5F 31 :B1


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5900 0A 17 F9 5A 17 F9 65 25 BA 1F 01 39 8D B5 17 F8 :72
5910 CD 9F 12 1F 10 17 F8 EE 17 F8 E3 17 F8 DE 17 F9 :99
5920 2A 81 0D 27 32 81 20 27 E5 81 1E 26 04 30 1E 20 :F5
5930 DD 81 4B 26 0C 17 F9 13 34 02 17 F8 CD 35 02 20 :67
5940 09 17 F9 18 24 04 30 1F 20 C4 A7 1F A1 1F 27 BE :F7
5950 86 07 17 F8 C9 20 EA 17 F8 84 16 F7 F2 CC AF A5 :21
5960 FD FF FA 03 18 10 DE 0A 7E AF A7 17 F8 FC DD 10 :D5
5970 16 F7 DC 16 F7 D9 17 FF 4A 17 F8 CF 81 0D 27 13 :D5
5980 81 20 10 26 FE 5C 34 10 17 FF 38 30 01 9F 12 35 :DA
5990 10 20 10 CC FF FF DD 12 20 09 17 FF 26 33 89 01 :1B
59A0 00 DF 12 86 12 17 F8 76 0F 1E 0C 1E 0A 1E 26 0F :C2
59B0 86 10 97 1E 34 10 30 8D 00 69 17 F8 2B 35 10 1F :53
59C0 10 17 F8 42 17 F8 37 DE 06 8D 30 34 01 17 F8 2E :BA
59D0 96 16 17 F8 35 17 F8 26 86 04 A7 C0 34 10 9E 06 :FE
59E0 17 F8 05 35 10 17 F7 F6 35 01 24 07 7D 03 13 27 :78
59F0 BB 9F 12 7F 03 13 0F 1E 16 F7 54 0F 16 8D 00 8D :CE
-----------------------------------------------------------
Sum: 0F BF 38 73 03 70 F3 C9 E7 C0 36 B9 86 C8 A2 03 :31


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5A00 00 17 F7 FA 8D 00 8D 00 9C 12 25 01 39 A6 84 81 :DA
5A10 20 24 02 86 2E A7 C0 A6 84 9B 16 97 16 17 F7 DC :D3
5A20 9C 12 39 0D 0A 00 00 00 41 44 44 52 20 20 2B 30 :B4
5A30 20 2B 31 20 2B 32 20 2B 33 20 20 2B 34 20 2B 35 :96
5A40 20 2B 36 20 2B 37 20 20 2B 38 20 2B 39 20 2B 41 :B6
5A50 20 2B 42 20 20 2B 43 20 2B 44 20 2B 45 20 2B 46 :EB
5A60 20 20 73 75 6D 0D 0A 00 00 00 04 20 16 17 FE 53 :4E
5A70 17 F7 8B 1F 10 17 F7 8E 17 F7 D0 81 59 10 26 F6 :48
5A80 CE AF 2A 17 00 98 10 DE 0A 35 FF 0C 1B 16 F6 BF :74
5A90 0F 1B 20 F9 73 5A 94 20 F4 17 FE 27 9F 0C 20 ED :AC
5AA0 17 F7 A8 DE 00 33 C8 2E 81 44 27 3F 34 02 17 FE :33
5AB0 12 35 02 81 43 27 1A 81 53 10 26 FD 25 EC C1 26 :4D
5AC0 FC 11 93 04 10 24 FD 1A AF 5E 6F C0 6F C4 16 FF :73
5AD0 BC EC C4 27 F9 AC C1 26 F8 34 40 EC C1 26 FC EC :46
5AE0 5C 35 10 ED 1E EC 5E ED 5C 20 E3 17 F6 F0 1F 31 :8F
5AF0 EC 84 27 DA 17 F7 03 20 F7 9E 02 30 02 17 F6 DE :56
-----------------------------------------------------------
Sum: 59 91 5B E2 AC 5E 76 99 CD 74 91 6E CB 65 60 5C :6C


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5B00 17 F6 F7 9C 04 26 F9 39 9E 02 86 3F 30 02 9C 04 :33
5B10 27 0B A1 94 26 F6 E6 88 10 E7 94 20 EF 39 9E 02 :64
5B20 EE 2A 86 3F 30 02 9C 04 27 12 11 A3 84 27 F5 A1 :DD
5B30 94 27 F1 E6 94 E7 88 10 A7 94 20 E8 39 8D BA 17 :7F
5B40 F6 9C 8D C4 9E 02 30 02 9C 04 10 27 F6 01 17 F6 :90
5B50 FA 81 0D 10 27 F5 F8 81 18 27 28 81 4F 27 14 17 :B6
5B60 F7 30 25 29 17 F6 F7 17 F7 02 25 D1 ED 84 17 F6 :FD
5B70 8D 20 D3 17 F6 D5 81 4E 26 04 8D A2 20 02 8D 88 :C1
5B80 16 F5 CC 6F 80 6F 80 9C 04 26 F8 20 B0 34 10 30 :B7
5B90 8C 07 17 F6 53 35 10 20 AD 20 20 20 20 04 50 1D :F6
5BA0 CC 4E 43 54 52 20 50 1E C2 59 54 45 20 20 50 1F :F4
5BB0 C3 48 4B 20 20 20 50 20 C3 4F 44 20 20 20 50 28 :54
5BC0 D3 55 42 52 53 20 50 2E C2 50 20 20 20 20 50 86 :15
5BD0 C2 50 45 20 20 20 50 96 C2 55 46 20 20 20 50 A6 :50
5BE0 D3 57 49 56 45 43 FF FA C9 4E 43 48 41 30 D3 36 :66
5BF0 CF 55 43 48 41 30 D3 52 CF 55 54 50 54 52 F9 35 :E1
-----------------------------------------------------------
Sum: 9C A2 25 52 FE 5E 45 C7 9F F6 E2 82 13 D7 24 74 :98
```



FM-8

BIOS&サブシステム

■広瀬仁彦

F-BASICでは，I/Oデバイスのアクセスを BIOS（Basic Input Output System）と呼ばれるモジュールが一括して行なっています．I/Oのアクセス・ルーチンを1箇所に集めてモジュール化する手法は，CP/MやFLEXなど，OSでは一般的な手法ですが，F-BASICでもこの手法を採り入れています．

BIOSモジュールの採用で，次のような効果が期待できます．

(1)F-BASICのために開発されたI/Oルーチン（BIOS）を，ほとんどそっくりそのまま，他のシステム（CP/MやFLEX，UCSD PASCAL，アプリケーション・プログラムなど）でも使える．

(2)I/Oデバイスのアクセス・ルーチンの開発，メンテナンスが一本化できるので，開発の効率がよく，ソフトウェアの信頼性も高くなる．

また，BIOSを採用した背景には，FM-8では，全アドレス空間64KBがRAMになるモードがあり，この場合，ROMがないので，ROMモニタを常駐させることができないということがあります．

以後の文章では，番地に"H"をつけていなくても16進表示であるものとします．

BIOSの概要

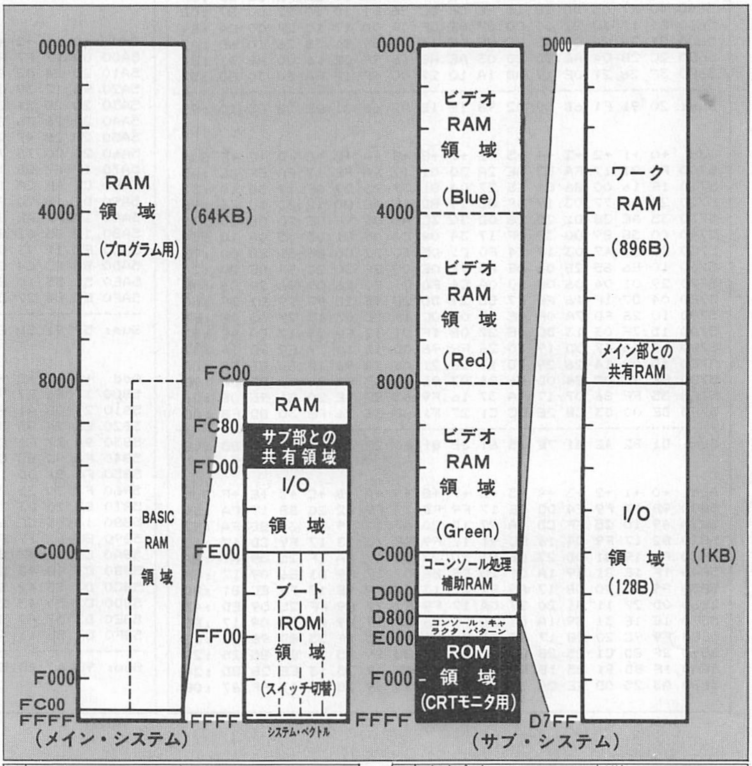
さて，BIOSが現在サポートしているI/Oは，次の9個です．

❶サブシステム（キーボードとCRTディスプレイ，タイマ）
❷ミニフロッピーディスク（最大4台）
❸プリンタ（ビット・イメージ）
❹磁気バブル・カセット（2ユニット）
❺ベル（連続音）
❻オーディオ・カセット（富士通独自方式）
❼アナログ・ポート（4チャンネル）
❽音声合成
❾漢字ROM（JIS第一水準）

BIOSとユーザーの情報交換は，RCB（Request Control Block）と呼ばれる8バイトのインターフェイス領域で行なわれます．BIOSのエントリ・ポイントは一つだけで，BIOSの中のどのI/Oルーチンを使用するかは，RCBの先頭バイトに設定するリクエスト番号により指定します．RCBの8バイトのうち先頭の2バイト（リクエスト番号とステータス）は常に共通ですが，残りの6バイトはリクエスト番号により，各バイトのもつ意味が決定されます．

現在，BIOSのI/Oルーチンは，28個用意されており，リクエスト番号は0番から27番までです．以下にリクエスト番号と処理の内容を示します（10進数で示す）．

FM-8メモリ・マップ



0	アナログ・ポート・データのリード
1	オーディオ・カセットのモータON/OFF
2	オーディオ・カセット1バイト・ライト
4	バブル・メモリ・イニシャライズ
5	CRT画面のハード・コピー（HARDC2）
6	バブル・カセットnページ・ライト
7	バブル・カセットnページ・リード
8	シーク・トラック・ゼロ
9	フロッピーディスク・セクタ・ライト
10	フロッピーディスク・セクタ・リード
11	音声合成スタート
12	ベルON
13	ベルOFF
14	プリンタ出力
15	CRT画面のハード・コピー（HARDC1）
16	サブシステム・アウトプット
17	サブシステム・インプット
18	サブシステム1行入力
19	サブシステム継続行入力
20	サブシステム1行出力
21	サブシステム1文字キー入力
22	漢字ROMリード
23	プリンタのステータス・チェック
24	BIOSイニシャライズ
25	音声合成パラメータの転送
26	音声合成データの転送
27	音声合成イニシャライズ

現在わかっているステータスの値の意味は次のとおりです（10進数で示す）．

0：ノーエラー

(a)BIOSシステム・エラー

1：RCBエラー
2：デバイス・アンアベイラブル・エラー

(b)フロッピーディスク・エラー

10：ドライブ・ノット・レディ
11：ディスク・ライト・プロテクト
12：ハード・エラー
　シーク・エラー
　ロスト・データ・エラー
　レコード・ノット・ファウンド・エラー
13：CRCエラー
14：デリーテッド・データ・マーク検出
15：タイム・オーバーエラー

(c)バブル・カセット・エラー

30：未定義コマンド

31：マーカ未検出
32：不良ループが多い
33：データ・トランスファーエラー
34：CRCエラー
35：ページ・アドレッシング・エラー
37：デバイス・ノット・レディ（バブル・カセット未装着）

(d)**プリンタ・エラー**
50：プリンタ・ペーパ・エンプティ
51：プリンタ・ノット・レディ

BIOSはメイン・システムのFC80H～FDFFHを主にアクセスして，I/Oデバイスをコントロールするモジュールであるわけです．

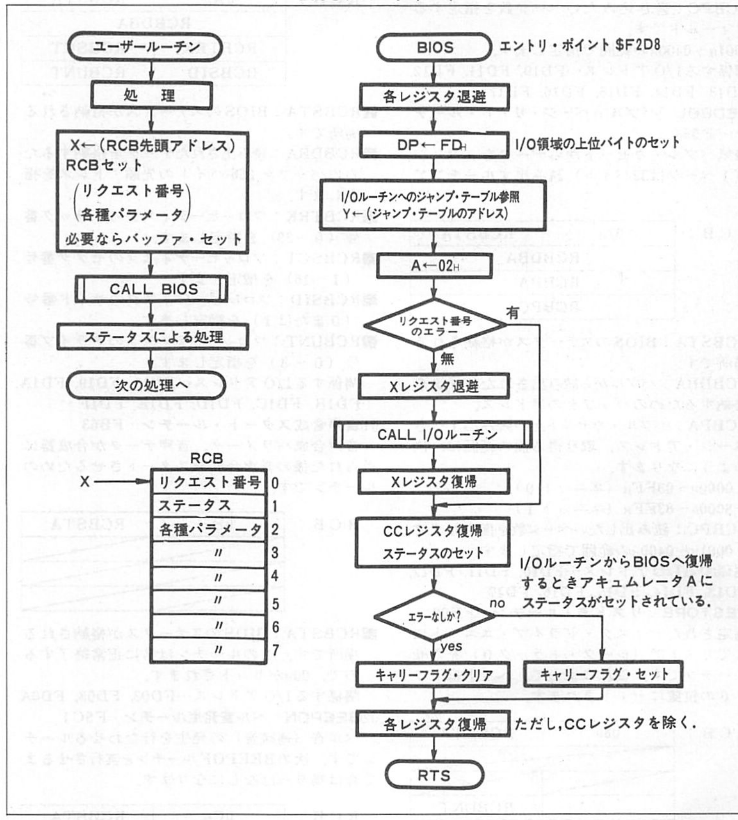
BIOSの使用法

さて，実際にBIOSを使ってみようとすると，決められた手続きをした上で，BIOSをコールしなければなりません．そこで，今度はBIOSの使い方を説明します．

BIOSをコールするにあたって，最低限必要なことは前にも述べたRCB（Request Control Block）と呼ばれる8バイトのインターフェイス領域のセットおよびCPUのXレジスタへRCBの先頭アドレスをセットすることの2点です．

これまでの説明でBIOS自体の使い方は，ある程度おわかり頂けたことと思います．しかし，RCBの各バイトの意味がわからなくては実際にBIOSを使うわけにはいかないので，現時点でわかったRCBの構成などを紹介したいと思います．

BIOSフローチャート



(0)**ANALGP　アナログ・ポート入力ルーチン……F33C**

4chあるアナログ・ポートのうち，指定したチャンネルを指定した電圧レンジで読み取る（8ビット）ルーチンです．

RCB：

00H	RCBSTA
RCBCNL	RCBRNG
RCBAPD	

■RCBSTA：BIOSのステータスの格納される場所です．
このルーチンは常に正常終了するので，00Hがセットされます．

■RCBCNL：チャンネル番号で，0から3の値をとります．入力したいチャンネルをこれによって選択してください．

■RCBRNG：電圧の範囲を指定するためのパラメータです．
"H"：0V～2.5Vの電圧範囲を指定します．
"L"：0V～0.625Vの電圧範囲を指定します．
このルーチンでは"H"のASCII CODE(48H)がここにセットされているかどうかで電圧レンジを切り換えています．

■RCBAPD：入力されたアナログ値に対するデジタル値がセットされます．
関係するI/Oアドレス…FD0C

(1)**オーディオ・カセットのモータON/OFFルーチン…F3FE**

オーディオ・カセットのモータを指定した状態（ON/OFF）にするルーチンです．

RCB：

01H	RCBSTA
RCBMOT	

■RCBSTA：BIOSのステータスの格納される場所です．
このルーチンは常に正常終了するので，00Hがセットされます．

■RCBMOT：モータの状態を指定するためのパラメータです．
FFH：モータONを指定します．
00H～FEH：モータOFFを指定します．
関係するI/Oアドレス…FD00

(2)**オーディオ・カセット1バイト・ライト・ルーチン…F40B**

オーディオ・カセットへ1バイト書き込みを行なうルーチンです．

FM-8では，オーディオ・カセットのデータの記録に1,600ボーのサッポロシティ・スタンダード方式を採用しています．

RCB：

02H	RCBSTA
RCBDAT	

■RCBSTA：BIOSのステータスが格納される場所です．このルーチンは常に正常終了するので，00Hがセットされます．

■RCBDAT：オーディオ・カセットへ書き込む1バイトのデータです．
関係するI/Oアドレス…FD00

(3)**オーディオ・カセット1バイト・リード・ルーチン…F484**

オーディオ・カセットからデータを1バイト読み出すルーチンです．

RCB：

03H	RCBSTA
RCBDAT	

■RCBSTA：BIOSのステータスが格納される場所です．

■RCBDAT：オーディオ・カセットから読み出した1バイトのデータが格納される場所です．
関係するI/Oアドレス…FD02

(4)**INTBBL　バブル・カセット　イニシャライズ・ルーチン…F511**

磁気バブル・カセットをイニシャライズするためのルーチンですが，ハードウェア的にバブルに対してリセットをかけるような作業を行なっています．

RCB：

04H	RCBSTA

■RCBSTA：BIOSのステータスが格納される場所です．
関係するI/Oアドレス…FD11, FD12, FD13

(5)**SCREEN　CRTスクリーン・イメージ・コピールーチン…F997**

CRTスクリーンの画面イメージをプリンタにコピーするルーチンです．このルーチンの機能はF-BASICのコマンドの**HARDC2**に相当します．CRT画面上のドットの色とプリンタ上のドットの色は，次のように対応します．

CRT画面上		プリンタ上
黒以外	→	黒
黒	→	白

RCB：

05H	RCBSTA
RCBBDA	

■RCBSTA：BIOSのステータスが格納される場所です．

■RCBBDA：作業用バッファ209バイトの先頭アドレスを指定します．
関係するI/Oアドレス…FD00, FD01, FD02, FD05

(6)**WRTBBL　バブルnページ・ライト・ルーチン…F553**

磁気バブル・カセットへデータをnページ分（1ページは32バイト）書き込むルーチンです．

RCB：

06H	RCBSTA
RCBDBA	
RCBPA	
RCBPC	

■RCBSTA：BIOSのステータスが格納される場所です．

■RCBDBA：バブル・カセットに書きこみを行うべきデータのバッファ・アドレスです．

■RCBPA：バブル・カセット上のページ・アドレスです．値の範囲は，次のようになります．
0000H～03FFH（ユニット0）

8000_H〜$83FF_H$（ユニット１）

■RCBPC：書き込みたいページ数を指定するフィールドです．

0001_H〜0400_Hの範囲で指定します．

関係するI/Oアドレス…FD10, FD11, FD12, FD13, FD14, FD15, FD16, FD17

(7)REDBBL　バブルｎページ・リード・ルーチン…F553

磁気バブル・カセットからデータをｎページ分（１ページは32バイト）読み出すルーチンです．

RCB：

07_H	RCBSTA
RCBDBA	
RCBPA	
RCBPC	

■RCBSTA：BIOSのステータスが格納される場所です．

■RCBDBA：バブルから読み出されたデータを格納するためのバッファのアドレス．

■RCBPA：バブル・カセット上の読み出すべきページ・アドレス．取り得る値の範囲は，次のようになります．

0000_H〜$03FF_H$（ユニット０）

8000_H〜$83FF_H$（ユニット１）

■RCBPC：読み出したいページ数を指定します．

0001_H〜0400_Hの範囲で指定します．

関係するI/Oアドレス…FD10, FD11, FD12, FD13, FD14, FD15, FD16, FD17

(8)RESTORE　リストア・ルーチン…FE02

指定されたディスク・ドライブ・ユニットに対してリストア（シーク・トラック０）をさせるルーチンです．これによって，ヘッドはトラック０の位置にセットされます．

RCB：

08_H	RCBSTA
	RCBUNT

■RCBSTA：BIOSのステータスが格納される場所です．

■RCBUNT：フロッピーディスク・ドライブ・ユニットの番号を指定します．ドライブ・ユニットの番号は０，１，２，３のいずれかを選択します．

関係するI/Oアドレス…FD18, FD1D, FD1F

(9)DWRITE　フロッピー１セクタ・ライト・ルーチン…FE05

フロッピーディスクに１セクタ（＝256バイト）書き込みを行なうルーチンです．F-BASICのレベルではセクタの指定は１〜32ですが，BIOSのレベルでは１〜16でセクタを指定すると同時に０，１でサイドの指定も行なって物理的位置を決定しています．

RCB：

09_H	RCBSTA
RCBBDA	
RCBTRK	RCBSCT
RCBSID	RCBUNT

■RCBSTA：BIOSのステータスが格納される場所です．

■RCBBDA：書き込みデータ・バッファ256バイトの先頭アドレスを設定します．

■RCBTRK：フロッピーディスクのトラック番号（０〜39）を指定します．

■RCBSCT：フロッピーディスクのセクタ番号（１〜16）を指定します．

■RCBSID：フロッピーディスクのサイド番号（０または１）を指定します．

■RCBUNT：フロッピーディスクのドライブ番号（０〜３）を指定します．

関係するI/Oアドレス…FD18, FD19, FD1A, FD1B, FD1C, FD1D, FD1E, FD1F

(10)DREAD　フロッピー１セクタ・リード・ルーチン…FE08

フロッピーディスクから１セクタ（＝256バイト）を読み出すルーチンです．

RCB：

$0A_H$	RCBSTA
RCBDBA	
RCBTRK	RCBSCT
RCBSID	RCBUNT

■RCBSTA：BIOSのステータスが格納される場所です．

■RCBDBA：読み出されたデータを格納するためのバッファ256バイトの先頭アドレスを指定します．

■RCBTRK：フロッピーディスクのトラック番号（０〜39）を指定します．

■RCBSCT：フロッピーディスクのセクタ番号（１〜16）を指定します．

■RCBSID：フロッピーディスクのサイド番号（０または１）を指定します．

■RCBUNT：フロッピーディスクのドライブ番号（０〜３）を指定します．

関係するI/Oアドレス…FD18, FD19, FD1A, FD1B, FD1C, FD1D, FD1E, FD1F

(11)音声合成スタート・ルーチン…FB63

音声合成パラメータ，音声データが合成器に送られた後の音声合成をスタートさせるためのルーチンです．

RCB：

$0B_H$	RCBSTA

■RCBSTA：BIOSのステータスが格納される場所です．このルーチンは常に正常終了するので，00_Hがセットされます．

関係するI/Oアドレス…FD03, FD08, FD0A

(12)BEEPON　ベル音発生ルーチン…F5C1

ベル音（連続音）の発生を行なわせるルーチンです．次のBEEPOFルーチンを実行させるまで音は鳴りっぱなしになります．

RCB：

$0C_H$	RCBSTA

■RCBSTA：BIOSのステータスが格納される場所です．このルーチンは常に正常終了するので，00_Hがセットされます．

関係するI/Oアドレス…FD03

なお，実験した結果このアドレスに偶数または00_H〜$3F_H$を書き込むと**BEEP0**，40_H〜$7F_H$のうちの奇数を書き込むと**BEEP**または**PRINT CHR$(7)**，$80_H$〜$FF_H$のうちの奇数を書き込むと**BEEP1**というF-BASICのコマンドに相当する機能が得られることがわかりました．

(13)BEEPOF　ベル音停止ルーチン…F5C4

BEEPONルーチンによって発生したベル音を停止させるルーチンです．

RCB：

$0D_H$	RCBSTA

■RCBSTA：BIOSのステータスが格納される場所です．このルーチンは常に正常終了するので，00_Hがセットされます．

関係するI/Oアドレス…FD03

(14)LPOUT　プリンタ出力ルーチン…F5CA

プリンタに文字を出力するためのルーチンです．

このルーチンは行という概念をもっていないので，改行を行なうためには，そのためのコードも印字データと共にバッファ内にセットして，プリンタに送ってやる必要があります．

出力の途中で，紙ぎれ，オフラインなどの理由によってプリンタがノット・レディの状態になり，その後再びレディ状態に戻ったような場合に，その行を打ち直すというような処置は行っていません．

RCB：

$0E_H$	RCBSTA
RCBDBA	
RCBLN	

■RCBSTA：BIOSのステータスが格納される場所です．

■RCBDBA：プリンタに出力したいデータを格納したバッファの先頭アドレスをここにセットしてください．

■RCBLN：データ・バッファの長さをここにセットしてください．

関係するI/Oアドレス…FD00, FD01, FD02

(15)HDCOPY　CRTスクリーン・ハード・コピー・ルーチン…F7D6

CRTスクリーンのイメージをプリンタにコピーするルーチンです．このルーチンは(5)SCREENルーチンとは異なり，スクリーン上の１ドットがプリンタ上では４ドットで表現されます．この結果，白灰黒の濃淡のついたプリント・イメージが得られます．

スクリーン上は８色のカラーが出力可能なので，各々の色と濃淡は１対１に対応しないので，RCRCDTによってその対応を指定します．このルーチンの機能はF-BASICのコマンドの**HARDC1**に相当します．

RCB：

$0F_H$	RCBSTA
RCBDBA	
RCBCDT	

■RCBSTA：BIOSのステータスが格納される場所です．

■RCBDBA：作業用バッファ210バイトの先頭アドレスを指定します．

■RCBCDT：上位バイトは，黒に対応する色を指定します．下位バイトは，灰に対応する色を指定します．

	7	6	5	4	3	2	1	0	
黒									RCBCDT上位バイト
灰									RCBCDT下位バイト
	白	黄	水色	緑	紫	赤	青	黒	

各色に対応したビットが１ならば，その色が指定の濃淡レベルでプリントされるようになっています．上位バイト，下位バイトで，両方同じビットが１になっている場合には，黒でプリントされます．

濃淡レベルは黒灰白の３通りになりますが，これらはプリンタのドット表現では次のようになります．

レベル	
黒	●●●●
灰	○●○●
白	○○○○

関係するI/Oアドレス…FD00, FD01, FD02, FD05

(16)サブシステム・アウトプット・ルーチン…F611

サブシステムに対してコマンドを出すときに使用するルーチンです．このルーチンは，レーズ・スクリーンなどのように，コマンドの実行結果が返される必要のない場合に使用します．

RCB：

10_H	RCBSTA
RCBDBA	
RCBLNT	

■RCBSTA：BIOSのステータスが格納される場所です．

■RCBDBA：サブシステムへ送るデータ・バッファ（最大128バイト）の先頭アドレスを指定します．

■RCBLNT：サブシステムへ送るデータ・バッファの長さをセットします．

関係するI/Oアドレス…FD05

⒄サブシステム・インプット・ルーチン…F611

サブシステムに対してコマンドを発行するときに使用するルーチンです．このルーチンは，キー入力のように，コマンドの実行結果が返されなければならない場合に使用します．

RCB：

11_H	RCBSTA
RCBDBA	
RCBLNT	
RCBLNR	

■RCBSTA：BIOSのステータスが格納される場所です．

■RCBDBA：サブシステムへの送受データ・バッファ（最大128バイト）の先頭アドレスを指定します．

■RCBLNT：サブシステムへ送るデータ・バッファの長さをセットします．このルーチンが終了するとき，ここにはRCBLNRの内容がコピーされています．

■RCBLNR：サブシステムから受け取るデータ・バッファの長さをセットします．

関係するI/Oアドレス…FD05

⒅サブシステム１行入力ルーチン…F675

サブシステムから１行分のデータを受け取るためのルーチンです．サブ部との共有領域は128バイトしかないため１行全部を受け取れなかった場合，ユーザーは次のサブシステム継続行ルーチンを使って，１行全部の情報を受け取ることができます．

RCB：

12_H	RCBSTA
RCBDBA	
RCBLNT	
RCBKNR	

■RCBSTA：BIOSのステータスが格納される場所です．

■RCBDBA：サブシステムへの送受データ・バッファの先頭アドレスを指定します．

■RCBLNT：サブシステムへ送るデータ・バッファの長さをセットします．このルーチンが終了する時，ここには１行入力データのバイト数がセットされています．

■RCBLNR：サブシステムから受け取るデータ・バッファの長さをセットします．

関係するI/Oアドレス…FD05

⒆サブシステム継続行入力ルーチン…F733

サブシステムから継続行のデータを受け取るためのルーチンです．サブ部との共有領域は128バイトしかないため，行の長さが長いときには，このルーチンを何回か使用する必要があります．

RCB：

13_H	RCBSTA
RCBDBA	
RCBLN	
RCBLNR	

■RCBSTA：BIOSのステータスが格納される場所です．

■RCBDBA：サブシステムから受け取るデータ・バッファの先頭アドレスを指定します．

■RCBLN：サブシステムからの継続行入力データのバイト数がセットされます．

■RCBLNR：サブシステムから受け取るデータ・バッファの長さをセットします．

関係するI/Oアドレス…FD05

⒇OUTPUT CRT出力ルーチン…F74D

RCBDBAによって示されるバッファ上のデータを，RCBLNによって指定されたバイト数分CRTに出力させるルーチンです．このルーチンでは表示色を変えたり（F-BASICのCOLOR文），表示開始位置を決める（F-BASICのLOCATE文）機能までは持っていません．

RCB：

14_H	RCBSTA
RCBDBA	
RCBLN	

■RCBSTA：BIOSのステータスが格納される場所です．

■RCBDBA：CRTへ出力させるデータ・バッファの先頭アドレスを指定します．

■RCBLN：データ・バッファの長さを指定します．

関係するI/Oアドレス…FD05

㉑KEYIN　キーボード１文字入力ルーチン…F7AD

キーボードから入力された文字のコードを受け取るルーチンです．

RCB：

15_H	RCBSTA
RCBDBA	
RCBLN	

■RCBSTA：BIOSのステータスが格納される場所です．

■RCBDBA：キー入力データ・バッファの先頭アドレスをセットします．バッファは２バイト用意すればよいだけです．RCBの８バイトの最後の２バイトが使われていないので，これをバッファとして用いればエリアの節約になります．

■RCBLN：キー入力データのバイト数を表しますが，常に0002_Hがセットされます．

キー入力データ・バッファ

KEY	EN

KEY：入力キーのコード

EN　：キー入力の有無

0：キー入力なし

1：キー入力あり

関係するI/Oアドレス…FD05

㉒漢字ROMリード・ルーチン…F37A

JISの漢字コードに対応する漢字のドット・パターン32バイトを読み出し，指定された領域に転送するルーチンです．

RCB：

16_H	RCBSTA
RCBDBA	
RCBJKC	

■RCBSTA：BIOSのステータスが格納される場所です．このルーチンは常に正常終了するので，00_Hがセットされます．

■RCBDBA：漢字ROMから読み出されたドット・パターン32バイトを格納するためのデータ・バッファの先頭アドレスを指定します．

■RCBJKC：漢字のJISコードをセットします．

関係するI/Oアドレス…FD20, FD21, FD22, FD23

㉓LPCUK　プリンタのレディチェック・ルーチン…F5E7

プリンタのレディチェックを行うルーチンです．

RCB：

17_H	RCBSTA

■RCBSTA：BIOSのステータスが格納される場所です．プリンタがレディであれば，正常終了を示すために00_Hがセットされます．エラーが発生した場合には，次のようなエラー番号がセットされます（10進数表示）．

50：ペーパーエンプティ

51：ノット・レディ

関係するI/Oアドレス…FD02

㉔BIOS　イニシャライズ・ルーチン…FB8A

現在のところ，このルーチンはベル音を停止させるだけのものです．

RCB：

18_H	RCBSTA

■RCBSTA：BIOSのステータスが格納される場所です．このルーチンは常に正常終了するので，00_Hがセットされます．

関係するI/Oアドレス…FD03

㉕音声合成パラメータの転送ルーチン…FAF7

音声合成を行なうには，まず音声合成パラメータを送らなければならないが，その転送を行なうのがこのルーチンです．

RCB：

19_H	RCBSTA
RCBPBA	
PCBLN	

■RCBSTA：BIOSのステータスが格納される場所です．このルーチンは常に正常終了するので，00_Hがセットされます．

■RCBPBA：音声合成パラメータが格納されているバッファの先頭アドレスを指定します．

■RCBLN：バッファの長さをセットします．

関係するI/Oアドレス…FD08, FD09, FD0A

㉖音声合成データの転送ルーチン…FB3A

音声合成を行なうには，まず音声合成パラメータを送り，つぎに音声データを合成器のバッファに送り，音声合成をスタートさせるという手順がいるが，音声データを合成器のバッファに送るのがこのルーチンの役割です．

RCB：

$1A_H$	RCBSTA
RCBDBA	
RCBLN	

■RCBSTA：BIOSのステータスが格納される場所です．

■RCBDBA：音声データが格納されているバッファの先頭アドレスを指定します．

■RCBLN：バッファの長さをセットします．

関係するI/Oアドレス…FD08, FD0B

㉗音声合成イニシャライズ・ルーチン…FB79

音声合成器をイニシャライズするルーチンです．

RCB：

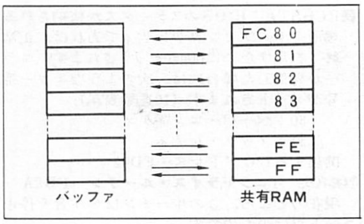
$1B_H$	RCBSTA

■RCBSTA：BIOSのステータスが格納される場所です．このルーチンは常に正常終了するので，00_Hがセットされます．

関係するI/Oアドレス…FD08, FD0A

サブシステムの使用法

サブシステムに対してコマンドを発行するルーチンは，BIOSの中に存在し，今回はリクエスト番号17（＝11_H）の“サブシステム・インプット・ルーチン”を使用して実験しました．



このルーチンではRCB中で指定するバッファ・アドレスから始まるバッファの内容をFC80番地から順に転送することによってコマンドを発行し，サブシステムに仕事をさせます．その後，サブシステムから返されるデータを同じバッファに転送します．従ってこのバッファの内容を適当な組合せにすれば，サブ・システムは働くはずです．

そこで各番地の意味を試行錯誤で調べてみたところ，FC82番地に書き込む数値がコマンド・コードであり，FC83番地以降へ書き込む数値はそのコマンドによって意味を持つパラメータとなります．FC80・FC81番地に書き込む数値の意味はわかりませんが，おそらくコマンドやパラメータを送ったことを，サブシステムへ知らせるためのものではないかと思われます．今回この2バイトはともに00Hとして実験しました．

(1)INIT

コンソール機能の初期化．

FC82：コマンド・コード01H
FC83：背景色コード（0～7）
FC84：横文字数（80または40）
FC85：縦行数（25または20）
FC86：スクロール開始行
FC87：スクロール行数
FC88：PF KEY表示スイッチ
FC89：画面消去スイッチ
FC8A：コンソール・カラースイッチ

F-BASICの**WIDTH, CONSOLE**文に相当します．**WIDTH, CONSOLE**文を実行すると画面消去も行なわれますが，文字数も行数も変更しなければ，画面消去スイッチをOFF(00H)にしてこのコマンドを実行すれば，画面はそのまま残すことができます．

(2)ERASE

画面およびコンソール・バッファを初期化．

FC82：コマンド・コード02H
FC83：消去範囲コード
　00：全画面
　01：スクロール画面
　02：ページ1画面
　03：ページ2画面
FC84：背景色コード
FC85：文字色コード

F-BASICの**CLS**文に相当します．

(3)PUT

文字列を表示する．

FC82：コマンド・コード03H
FC83：表示文字数
FC84：表示文字列
　⋮　　⋮

ただし，次のような特別の意味を持つ文字列もあります．

11H	文字色コード	COLOR文の機能	
12H	X	Y	LOCATE文の機能
13H	繰返し数	表示文字コード	STRING$関数の機能

このコマンドに対してはBIOSでI/Oルーチンが用意されており，リクエスト番号20(=14H)の"サブシステム1行出力ルーチン"を使用すればよい．F-BASICの**PRINT**文に相当します．

(4)GET

オペレータによる入力を可能にし，変更された最初のフィールドを取り込む．

FC82：コマンド・コード04H
FC83：表示文字数
FC84：(出力) 表示文字列
　　(入力) 変更文字数
FC85：(出力) 表示文字列
　　(入力) 変更文字列
　⋮　　⋮

このコマンドに対してはBIOSでI/Oルーチンが用意されており，リクエスト番号18(=12H)の"サブシステム1行入力ルーチン"を使用すればよい．F-BASICの**INPUT**"プロンプト・メッセージ"に相当します．

(5)GETC

GET後の未転送の変更されたフィールドを取り込む．FC82：コマンド・コード05Hと思われる．

このコマンドに対してはBIOSでI/Oルーチンが用意されており，リクエスト番号19(=13H)の"サブシステム継続行入力ルーチン"を使用すればよい．

(6)GET Character Block 1

指定枠内の文字コードを取り込む．

FC82：コマンド・コード06H
FC83：(出力) 横方向キャラクタ座標X_1
　　(入力) 入力文字数
FC84：(出力) 縦方向キャラクタ座標Y_1
　　(入力) 入力文字列
FC85：(出力) 横方向キャラクタ座標X_2
　　(入力) 入力文字列
FC86：(出力) 縦方向キャラクタ座標Y_2
　　(入力) 入力文字列
FC87：(出力) 入力文字列
　⋮　　⋮

F-BASICの**GET@**文形式1に相当します．

(7)PUT Character Block 1

指定枠内へ文字を表示する．

FC82：コマンド・コード07H
FC83：横方向キャラクタ座標X_1
FC84：縦方向キャラクタ座標Y_1
FC85：横方向キャラクタ座標X_2
FC86：縦方向キャラクタ座標Y_2
FC87：表示文字色
FC88：表示文字数
FC89：表示文字列
　⋮　　⋮

F-BASICの**PUT@**文形式1に相当します．

(8)GET Character Block 2

指定枠内の文字コードとアトリビュート文字を取り込む．

FC82：コマンド・コード08H
FC83：(出力) 横方向キャラクタ座標X_1
　　(入力) アトリビュート文字数＋入力文字数
FC84：(出力) 縦方向キャラクタ座標Y_1
　　(入力) アトリビュート文字
FC85：(出力) 横方向キャラクタ座標X_2
　　(入力) 入力文字
FC8686：(出力) 縦方向キャラクタ座標Y_2
　　(入力) アトリビュート文字
FC87：(入力) 入力文字
FC88：(入力) アトリビュート文字
FC89：(入力) 入力文字
　⋮　　⋮

F-BASICの**GET@**文形式2に相当します．

(9)PUT Character Block 2

指定枠内へアトリビュート付きの文字を表示する．

FC82：コマンド・コード09H
FC83：横方向キャラクタ座標X_1
FC84：縦方向キャラクタ座標Y_1
FC85：横方向キャラクタ座標X_2
FC86：縦方向キャラクタ座標Y_2
FC87：未使用
FC88：アトリビュート文字数＋表示文字数
FC89：アトリビュート文字
FC8A：表示文字コード
　⋮　　⋮

F-BASICの**PUT@**文形式2に相当します．

(10)GET Buffer Address

現在のコンソール・バッファのバッファ・ポインタの値を取り込む．

FC82：コマンド・コード0AH
FC83：(入力) 横方向キャラクタ座標X
FC84：(入力) 縦方向キャラクタ座標Y

(11)TABSET

タブ停止位置を設定する．

FC82：コマンド・コード0BH

FC83	タブ停止位置(ビットONの位置)	0～7
FC84	〃	8～15
FC85	〃	16～23
FC86	〃	24～31
FC87	〃	32～39
FC88	〃	40～47
FC89	〃	48～55
FC8A	〃	56～63
FC8B	〃	64～71
FC8C	〃	72～79

(12)CONSOLE Control

種々のコンソール機能の選択．

FC82：コマンド・コード0CH
FC83：コントロール・コード

各ビットの持つ意味は次の通りである．

FM-8のサブ・システム

●編集部

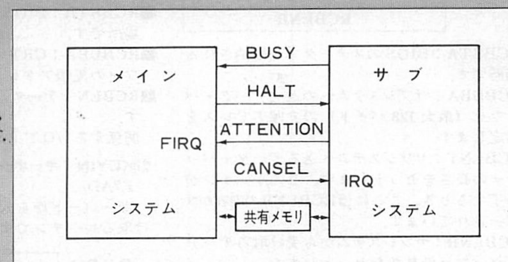
FM-8では，CRTグラフィック処理専用にサブCPUが使われています．この部分をサブ・システムと呼びます．

メイン・システムとサブシステムの通信はFC80番地から128バイトの共有メモリを介して行ないます．この通信は，BIOSのシステム・インプット/アウトプットを使えば，自動的に行なってくれます．

BIOSにたよらずサブシステムと通信をすることもできます．

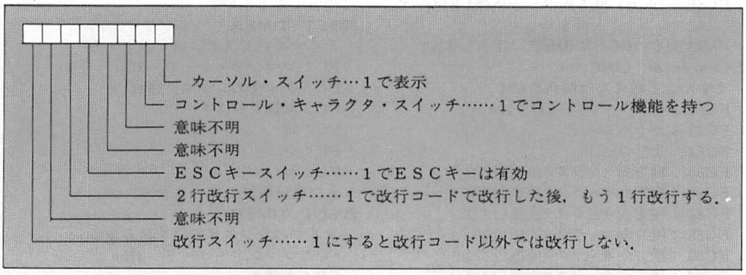
右記の制御フラグを用いて，同時に共有メモリをアクセスしないようにします．

サブ システムがBUSY	$FD05のビット7がオン(Read)
サブ システムをHALT	$FD05のビット7をオン(Write)
サブ システムをSTART	$FD05のビット7をオフ(Write)
サブ システムからATTENTION	$FD04のビット0がオフ(Read)
サブ システムをABORT	$FD05のビット6をオン(Write)



◘参考文献

渡部，森，"FM-8のハードウェア概説"，インターフェース，1981.9



詳しく説明すると，

・カーソル・スイッチ
　1……カーソルを表示する.
　0……カーソルを表示しない.

・コントロール・キャラクタ・スイッチ
　1……コントロール・キャラクタのコントロール機能を有効にする.
　0……コントロール・キャラクタのコントロール機能を無効とし，文字として表示する.

・ESCキースイッチ
　1…ESCキーを有効にし，ESCキーによる1行毎の表示停止が行なえる.
　0…ESCキーを無効とする.

・2行改行スイッチ
　1…行（改行コードまで）と行との間に1行入れる.
　0…行と行とをつめる.

・改行スイッチ
　1…改行コード以外では改行しない．すなわち横の文字数を例えば80字とすると，81字以上にわたる行は，改行せず同じ場所に表示される.
　0…改行コード以外でも横の文字数をこえる行は，そのつど改行する.

⒀Clear Screen

画面の初期化

FC82：コマンド・コード　$0D_H$
FC83：消去範囲コード
　00：全画面
　01：スクロール画面
　02：ページ1画面
　03：ページ2画面
FC84：背景色コード
FC85：文字色コード

F-BASICの**CLS**文に相当します．現在，解読した限りでは“ERASE”と同じです．F-BASICの**CLS**文はこちらのコマンドを使用しています.

⒁LINE

線または四角を描く.

FC82：コマンド・コード　15_H
FC83：カラーコード
FC84：機能コード
　00：PSET
　01：PRESET
　02：OR
　03：AND
　04：XOR
FC85：横方向ドット座標（上位）}
FC86：横方向ドット座標（下位）} X_1
FC87：縦方向ドット座標（上位）}
FC88：縦方向ドット座標（下位）} Y_1
FC89：横方向ドット座標（上位）}
FC8A：横方向ドット座標（下位）} X_2
FC8B：縦方向ドット座標（上位）}
FC8B：縦方向ドット座標（下位）} Y_2
FC8C：オプション・コード
FC8D：
　00：オプションなし
　01：Bオプション
　02：BFオプション

F-BASICの**LINE**文形式2に相当します.

⒂CHAIN

指示座標間を直線で結ぶ.

FC82：コマンド・コード　16_H
FC83：カラーコード
FC84：機能コード
　00：PSET
　01：PRESET
　02：OR
　03：AND
　04：XOR
FC85：指示座標数
FC86：横方向ドット（上位）}
FC87：横方向ドット（下位）} X_1
FC88：縦方向ドット（上位）}
FC89：縦方向ドット（下位）} Y_1
FC8A：横方向ドット（上位）}
FC8B：横方向ドット（下位）} X_2
FC8C：縦方向ドット（上位）}
FC8D：縦方向ドット（下位）} Y_2
FC8E：横方向ドット（上位）}
FC8F：横方向ドット（下位）} X_3
FC90：縦方向ドット（上位）}
FC91：縦方向ドット（下位）} Y_3

F-BASICの**CONNECT**文に相当します.

⒃POINT 横

点を表示します.

FC82：コマンド・コード　17_H
FC83：表示点数.
FC84：横方向ドット座標（上位）}
FC85：横方向ドット座標（下位）} X_1
FC86：縦方向ドット座標（上位）}
FC87：縦方向ドット座標（下位）} Y_1
FC88：カラーコード1
FC89：機能コード1
　00：PSET
　01：PRESET
　02：OR
　03：AND
　04：XOR
FC8A：横方向ドット座標（上位）}
FC8B：横方向ドット座標（下位）} X_2
FC8C：縦方向ドット座標（上位）}
FC8D：縦方向ドット座標（下位）} Y_2
FC8E：カラーコード2
FC8F：機能コード2
　⋮　　⋮

F-BASICの**PSET**文，または**PRESET**文に相当します.

⒄PAINT

指示座標から境界色（複数）の内部をぬる.

FC82：コマンド・コード　18_H
FC83：横方向ドット座標（上位）}
FC84：横方向ドット座標（下位）} X
FC85：縦方向ドット座標（上位）}
FC86：縦方向ドット座標（下位）} Y
FC87：カラーコード
FC88：境界色数
FC89：境界色1
FC8A：境界色2
FC8B：境界色3
　⋮　　⋮

F-BASICの**PAINT**文に相当します.

⒅SYMBOL

文字列を指示座標より，指示角度（90°単位），大きさで表示する.

FC82：コマンド・コード　19_H
FC83：カラーコード
FC84：機能コード
　00：PSET
　01：PRESET
　02：OR
　03：AND
　04：XOR
　05：NOT
FC85：角度コード
　00：ノーマル
　01：90°左回転
　02：180°左回転
　03：270°左回転
FC86：横倍率
FC87：縦倍率
FC88：横方向ドット座標（上位）}
FC89：横方向ドット座標（下位）} X
FC8A：縦方向ドット座標（上位）}
FC8B：縦方向ドット座標（下位）} Y
FC8C：表示文字数
FC8D：表示文字列
　⋮

F-BASICの**SYMBOL**文に相当します.

⒆Change COLOR

指定枠内の色を任意の組合せで変える.

FC82：コマンド・コード　$1A_H$
FC83：横方向ドット座標（上位）}
FC84：横方向ドット座標（下位）} X_1
FC85：縦方向ドット座標（上位）}
FC86：縦方向ドット座標（下位）} Y_1
FC87：横方向ドット座標（上位）}
FC88：横方向ドット座標（下位）} Y_2
FC89：縦方向ドット座標（上位）}
FC8A：縦方向ドット座標（下位）} Y_2
FC8B：変化色数
FC8C：カラーコード1
FC8D：カラーコード1'
FC8E：カラーコード2
FC8F：カラーコード2'
FC90：カラーコード3
FC91：カラーコード3

カラーコード1→カラーコード1'のように，指定色を別の色にすることができます．枠の指定方法は**LINE**などと同様です.

⒇GET Block 1

指定枠内の指定色（複数）を読み取る.

FC82：コマンド・コード　$1B_H$
FC83：（出力）横方向ドット座標（上位）
　　（入力）データ数
FC84：（出力）横方向ドット座標（下位）
　　（入力）データ列　} X_1
FC85：（出力）縦方向ドット座標（上位）
　　（入力）データ列
FC86：（出力）縦方向ドット座標（下位）
　　（入力）データ列　} Y_1
FC87：（出力）横方向ドット座標（上位）
　　（入力）データ列
FC88：（出力）横方向ドット座標（下位）
　　（入力）データ列　} X_2
FC89：（出力）縦方向ドット座標（上位）
　　（入力）データ列
FC8A：（出力）縦方向ドット座標（下位）
　　（入力）データ列　} Y_2
FC8B：（出力）指定色数
　　（入力）データ列
FC8C：（出力）指定色1
　　（入力）データ列
FC8D：（出力）指定色2
　　（入力）データ列
　⋮　　⋮

F-BASICの**GET@**文形式3に相当します.

(21)PUT Block 1

指定枠内へ指定色でドットを表示する.

FC82：コマンド・コード　$1C_H$
FC83：横方向ドット座標（上位）}
FC84：横方向ドット座標（下位）}
FC85：縦方向ドット座標（上位）}
FC86：縦方向ドット座標（下位）}
FC87：横方向ドット座標（上位）}
FC89：横方向ドット座標（下位）}
FC89：縦方向ドット座標（上位）}
FC8A：縦方向ドット座標（下位）}
FC8B：カラーコード
FC8C：機能コード
　00：PSET
　01：PRESET
　02：OR

03：AND
04：XOR
05：NOT
FC8D：データ数
FC8E：データ列
：　　：

F-BASICの**PUT@**文形式3に相当します.

(22)GET Block 2

指定枠内の3原色を読み取る.

FC82：コマンド・コード　$1D_H$
FC83：(出力) 横方向ドット座標 (上位)
　　　(入力)
FC84：(出力) 横方向ドット座標 (下位)
　　　(入力)　｝X_1
FC85：(出力) 縦方向ドット座標 (上位)
　　　(入力)
FC86：(出力) 縦方向ドット座標 (下位)
　　　(入力)　｝Y_1
FC87：(出力) 横方向ドット座標 (上位)
　　　(入力)
FC88：(出力) 横方向ドット座標 (下位)
　　　(入力)　｝X_2
FC89：(出力) 縦方向ドット座標 (上位)
　　　(入力)
FC8A：(出力) 縦方向ドット座標 (下位)　｝Y_2
　　　(入力) データ列
FC8B：(入力) データ列
FC8C：(入力) データ列
　：

データ列はBlue…Red…Green…のように, 各色毎にまとまって返されます. F-BASIC の**GET@**文形式4に相当します.

(23)PUT Block 2

GET Block 2で読んだデータを表示する

FC82：コマンド・コード　$1E_H$
FC83：横方向ドット座標 (上位)
FC84：横方向ドット座標 (下位)　｝X_1
FC85：縦方向ドット座標 (上位)
FC86：縦方向ドット座標 (下位)　｝Y_1
FC87：横方向ドット座標 (上位)
FC88：横方向ドット座標 (下位)　｝X_2
FC89：縦方向ドット座標 (上位)
FC8A：縦方向ドット座標 (下位)　｝Y_2
FC8B：未使用
FC8C：機能コード
　　00：PSET
　　01：PRESET
　　02：OR
　　03：AND
　　04：XOR
　　05：NOT
FC8D：データ数
FC8E：データ列 (Blue)
　　　：
　　データ列 (Red)
　　　：
　　データ列 (Green)
　　　：

F-BASICの**PUT@**文形式4に相当します.

(24)Graphic CURSOR

オペレータの指示した座標 (複数) を読み取る.

FC82：コマンド・コード　$1F_H$
FC83：(出力) カラーコード
　　　(入力) 横方向ドット座標(上位)
FC84：(出力) 入力座標数
　　　(入力) 横方向ドット座標(下位)　｝X_1
FC85：(出力) 横方向ドット座標(上位)
　　　(入力) 縦方向ドット座標(上位)
FC86：(出力) 横方向ドット座標(下位)
　　　(入力) 縦方向ドット座標(下)　｝Y_1
FC87：(出力) 縦方向ドット座標(上)
　　　(入力) 横方向ドット座標(下)
FC88：(出力) 縦方向ドット座標(上)
　　　(入力) 横方向ドット座標(下)　｝X_2
FC89：(入力) 縦方向ドット座標(上)
FC8A　(入力) 縦方向ドット座標(下)　｝Y_2
FC8B　(入力) 横方向ドット座標(上)
FC8C　(入力) 横方向ドット座標(下)　｝X_3
FC8D　(入力) 縦方向ドット座標(上)
FC8E：(入力) 縦方向ドット座標(下位)　｝Y_3
　：

F-BASICの**GCURSOR**文に相当します.

(25)Character LINE

文字による線または四角を描く.

FC82：コマンド・コード　20_H
FC83：カラーコード
FC84：文字コード
FC85：横方向キャラクタ座標(上位)
FC86：横方向キャラクタ座標(下位)　｝X_1
FC87：縦方向キャラクタ座標(上位)
FC88：縦方向キャラクタ座標(下位)　｝Y_1
FC89：横方向キャラクタ座標(上位)
FC8A：横方向キャラクタ座標(下位)　｝X_2
FC8B：縦方向キャラクタ座標(上位)
FC8C：縦方向キャラクタ座標(下位)　｝Y_2
FC8D：オプション・コード
　　00：オプションなし
　　01：Bオプション
　　02：BFオプション

F-BASICの**LINE**文形式1に相当します.

(26)INKEY

キーコードを読み取る.

FC82：コマンド・コード　29_H
FC83：(出力) Nullスイッチ
　　01…INPUT$(1)の機能
　　00…INKEY$の機能
　　(入力) 入力文字コード
FC84：(入力) KEY入力フラグ
　　01…KEY入力有
　　00…KEY入力無
　：

このコマンドに対してはBIOSでI/Oルーチンが用意されており, リクエスト番号21(=15_H)の"サブシステム1文字キー入力ルーチン"を使用すればよい. F-BASICの**INKEY$**もしくは**INPUT$(1)**に相当します.

(27)DEF PF

ファンクション・キーに文字列を定義する.

FC82：コマンド・コード　$2A_H$
FC83：プログラマブル・ファンクションキー番号
FC84：文字数
FC85：文字列
　：　　：

F-BASICの**KEY**文 (文字列定義) に相当します.

(28)GET PF

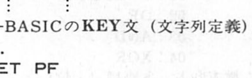
ファンクション・キーに定義されている文字列を読み取る.

FC82：コマンド・コード　$2B_H$
FC83：プログラマブル・ファンクション・キー番号
FC84：(入力) 文字数
FC85：(入力) 文字列
　：　　：

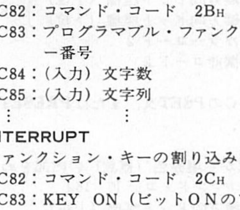
(29)INTERRUPT

ファンクション・キーの割り込みを選択する.

FC82：コマンド・コード　$2C_H$
FC83：KEY ON (ビットONのファンクション・キー)



FC84：KEY ON (ビットONのファンクション・キー)



CONSOLE文によってファンクション・キーを表示すると, KEY ONのファンクション・キーの色が変わるため割り込みが選択されている状態がわかります. F-BASICの**KEY(n) ON/OFF**に相当するようです.

(30)SET TIMER

タイマのレジスタに値を設定する.

FC：コマンド・コード　$3D_H$
FC：タイマのレジスタ選択.
FC：制御レジスタ
FC：時
FC：分
FC：秒

まだ充分解読していません.

(31)GET TIMER

タイマのレジスタの値を読み取る.

FC：コマンド・コード　$3E_H$
FC：未使用
FC：制御レジスタ
FC：(入力) 時
FC：(入力) 分
FC：(入力) 秒

まだ充分解読していません.

おわりに

以上がサブシステムに関してわかったことです. これだけでも, 機械語でゲームを作ろうという人には大いに役立つものと信じています. 試行錯誤で実験を繰り返したため, 間違いなどがあるかも知れませんので間違いを発見したり, 新事実を見出した方は是非ともお知らせ願いたいと思っております. FM-8をもっと使いこなそうじゃありませんか‼

注意

今後ソフトウェアの改良などにより, BIOSおよび各I/Oルーチンのエントリ・アドレス, その他が変わることがあります.

私が調べた所では, FM-8のBASIC ROMおよびCRTモニタROMには, かなりいくつかのバージョンがあり, 8900_H～8909_Hの10バイトにその情報 (内部バージョン, 日付) が格納されているようです.

1.02～1.05までの4種類があることは確認しています. 1.00, 1.01が市場に出ているかどうかはわかりませんが, たとえば, 1.03と1.05とを比較してみると前者は**BEEP**がSyntax Errorになるのに対し, 後者はなりません. また, **PAINT**文は後者の方が5～6倍も速いなどの違いがあります.

そのため, 今後FM-8の内部ルーチンを使うような記事には是非ともシリアル・ナンバーと, 8900_Hからの10バイトの内容も示して欲しいと思います. 参考までに, 今回この原稿を書くのに使った時点では, FM-8のROMは1.03で, 現在このFM-8 (大学の研究室のですが) は我が家と同じ1.05になっています.

◘参考資料

1) "F-BASIC文法書", 富士通株式会社
2) "FM-8のソフトウェア概説[下]", インターフェース, '81年11月号, CQ出版社
3) "FM-8のBIOS活用法と開発ツールへの応用", インターフェース, '82年2月号, CQ出版社
4) F-BASIC ROM version 1.03逆アセンブラ・リスト, FUJITSU MICRO 8 S/N 381080595

F-BASICに Change COLORコマンドを！

FM-8は優れたグラフィック機能を有し，SYMBOL文などは非常に重宝なコマンドです．また PAINT文の速さや，メモリ不足を生じない点など他の機種にも負けず，むしろ数段上をいっている感じさえします．こんなFM-8にも，まだF-BASICでサポートされていないグラフィック機能—それがChange COLORコマンドなのです．

このコマンドの機能がどのようなものなのかを一言で言うならば，

指定枠内の色を任意の組合せで変える

ということになります．

つまり長方形の領域を指定すると，その領域内の色を好きなように変えることができるのです．もちろん1色だけでも構いませんし，8色とも別の組合せにすることもできます．

こんなおもしろい機能があるにもかかわらず，F-BASICでサポートされていないためかCRTモニタROMのバージョン・アップが行なわれても，この機能についてのソフトウェアの改良は行なわれていないのが現状のようです．PAINTが前のバージョンから今のバージョン1.03になって5～6倍もスピード・アップしたのに比較すると大違いです．

そこでこのChange COLORコマンドを，F-BASICで使えるようにして見ようというわけです．そのため次の2つのことを満足できるものにします．

①DISK BASICでも使えること

②各パラメータは省略することも可能であること

この2つを満足するため，まずキーワードはROM上に存在するものを使い，このJump Tableを書きかえることにします．

ROM上にあるキーワードのうちChange COLORに最もふさわしいのは，やはりCOLORです．そこでCOLOR@をChange COLORのコマンドにすることに決めました．PC-8801にも，COLOR@文があるようですが，内容的にはまったく異なっているのでご注意を！

次に書式を決めるわけですが，まず枠の指定方法はLINE文などに準じることにし，次に何色を何色にするかの指定はカッコでくくった左側のコードの色を右側のコードの色にするということにします．

このようにして決めたCOLOR@文の書式は次のようになります．

COLOR@〔〔(X_1, Y_1)〕－〔(X_2, Y_2)〕〕〔，(カラー1，カラー1')〔，(カラー2，カラー2')〔……〔，(カラー8，カラー8')〕…〕〕〕

a)始点位置，終点位置の一方もしくは両方を省略することができる．

ただし一方のみを省略した場合には，どちらを省略したのかがわかるように“－”は必ず残しておくこと．また始点のみを省略した場合には前の終点が今回の始点となります．

b)カラーコードの組み合せの省略ができる．

打ち込み方としてはCLEAR，&H5FFFを実行した後，6280H番地から順に入れて62A4H番地をコールしてください．このとき6000H番地からJump Tableが入っていますので誤って書きかえないように！

6809をはじめてまだ半年も経過していないので，まだムダが多いのですが何とか正常に走るみたいです．ON ERROR GOTO文でも正常に働いてくれるので，いまのところバグはなさそうです．

ダンプ・リスト（F-BASICにchange COLORコマンドを）

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6280  10 00 62 88 00 0E 00 00 00 00 1A 00 0A 00 0A 00 :36
6290  64 00 64 01 00 02 01 01 02 02 03 03 04 04 05 05 :E9
62A0  06 06 00 02 34 7F BE 01 F2 CE 60 00 10 8E 01 40 :7F
62B0  EC 81 ED C1 31 3F 26 F8 CC 62 C6 FD 60 6A CC 60 :90
62C0  00 FD 01 F2 35 FF F6 01 E5 81 40 10 26 68 37 9D :33
62D0  D2 10 27 00 BA 81 2C 27 67 81 DA 26 10 FC 62 8F :7C
62E0  FD 62 8B FC 62 91 FD 62 8D 9D D2 20 2E BD 97 2F :05
62F0  BD 9E 8F 8C 02 80 10 24 38 09 BF 62 8B BD 97 32 :9F
-----------------------------------------------------------
Sum:  F2 94 F5 C6 B8 5F 14 A8 D1 DA EE B8 6D DA A3 32 :81
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6300  BD 9E 8F 8C 00 C8 10 24 37 F9 BF 62 8D BD 97 2C :D0
6310  C6 DA BD 97 38 27 78 81 2C 27 25 BD 97 2F BD 9E :A2
6320  8F 8C 02 80 10 24 37 DB BF 62 8F BD 97 32 BD 9E :74
6330  8F 8C 00 C8 10 24 37 CB BF 62 91 BD 97 2C 27 4F :C1
6340  7F 62 93 BD 97 32 BD 97 2F BD 9E 49 C1 08 10 24 :1E
6350  37 B1 F7 62 A2 BD 97 32 BD 9E 49 C1 08 10 24 37 :41
6360  A2 F7 62 A3 8E 62 94 B6 62 93 48 30 86 B6 62 A2 :85
6370  ED 84 F1 62 A2 26 03 7A 62 93 7C 62 93 BD 97 2C :EF
6380  27 0D 86 08 B1 62 93 22 BA 9D D8 10 26 33 B1 F6 :C9
6390  62 93 27 0B 58 CB 0C F7 62 85 BE 62 80 0E DE 39 :C9
-----------------------------------------------------------
Sum:  6F BE D8 A2 CA DB 80 5D AD 87 15 A7 DA 16 F4 0F :0C
```

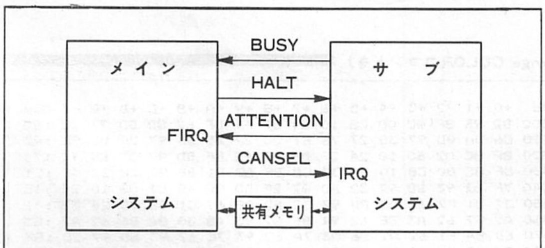
サブシステム使用法

■COMPAC

FM-8では，CRTグラフィック処理，キー入力処理，タイマ処理専用にサブCPUが使われています．このサブシステムを上手に使わないと，高速ゲームなどは作れません．

メイン・システムとサブシステム間の情報交換は，通信用の共有メモリ(128バイト）と４本の制御線で行なわれます．

図１　メインとサブの情報交換



1. 共有メモリ

メイン・システム側のアドレス＄FC80～＄FCFF(128バイト）と，サブシステム側のアドレス＄D380～＄D3FFの領域は，メインとサブで共有する領域です．この領域をお互いに同時にアクセスしないようにして，データの受け渡しをします(図２)．

2. 制御線

共有メモリを同時にアクセスしないようにするには，４本の制御線を使います．

メイン・システムからサブシステムをアクセスするには，図３のようにします．このようにして，サブシステムをHALTさせた後，共有メモリにサブシステムへのコマンドを書き込みます．

共有メモリに書き込んだ後は，サブシステムを図４のようにして，HALT解除します．

2-1 サブからメインへのBUSY信号…$FD05のビット７をREAD

サブシステムは，メイン側にBUSY信号（１：命令実行中，０：実行終了）を送ります．サブ側がBUSYの間は，メイン側で共有メモリをアクセスしないようにします．

2-2 メインからサブへのHALT命令信号……$FD05のビット７をWRITE

メイン・システムは，サブCPUをHALT(停止）させたり，HALT解除（開始）させることができます（１：HALT，０：解除)．

2-3 メインからサブへの割り込み信号……＄FD05のビット６をON

サブシステムが命令を実行中であっても，メイン・システムは，サブCPUにIRQ割り込みを発生させ，強制的に命令を中断させることができます．

2-4 サブからメインへの割り込み信号……＄FD04のビット０がON

サブシステムは，メイン側にFIRQ割り込みを発生させ，インターバル・タイマ，クロック，プログラマブル・ファンクション・キーの動作を知らせます．メイン・システムのFIRQ割り込みには，STOPキーなどもあるので，区別するにはこの信号を調べます．

割り込みを受けたことが判ったなら，サブCPUをHALTさせ，共有メモリに書かれているステータスをみて，何の割り込みか調べます．調べ終ったなら，＄FC80のビット７をONにしてから，サブCPUのHALTを解除します．

3. サブシステム入出力

サブシステムへBIOSを介さず直接コマンドを送れます．コマンドのフォーマットについては，ディスプレイ・サブシステム解説を参照してください．

3-1 サブシステム出力

サブシステムへの出力は非常に簡単で，図５のようになります．

3-2 サブシステム入力

サブシステムにコマンドを送り，サブシステムから応答をもらう場合があります．このときは，やや複雑になり，図６のようになります．＄FC80のビット７オンは，サブシステムをリセット（開始）する際，何もコマンドが設定されてないことを示すものです．

図2　データ受け渡し概念　フローチャート

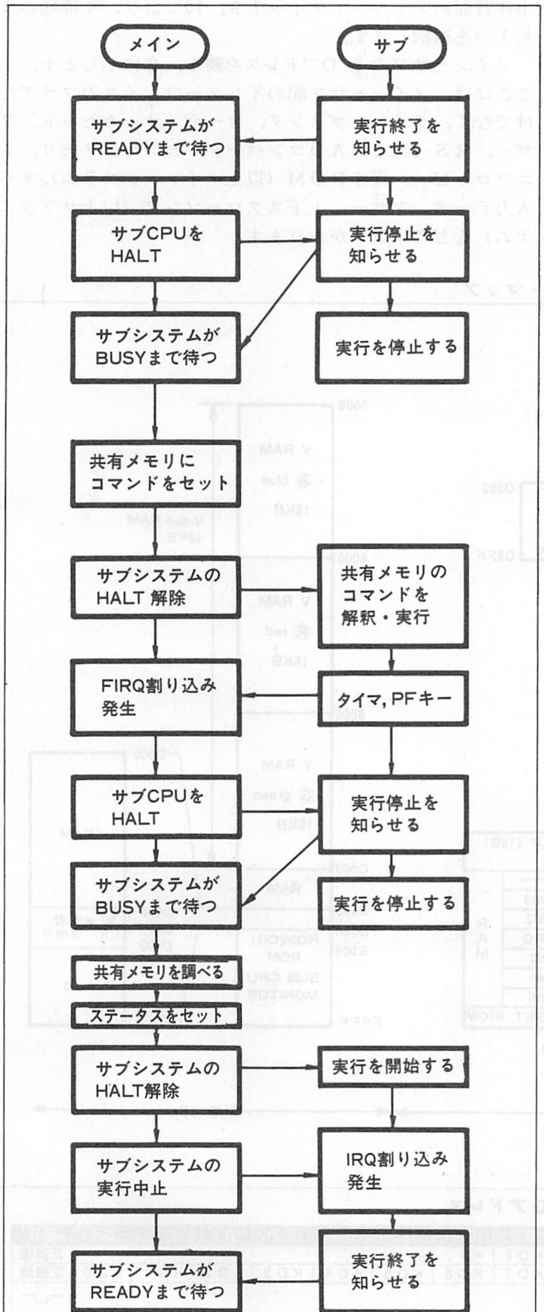


図3　サブCPU停止

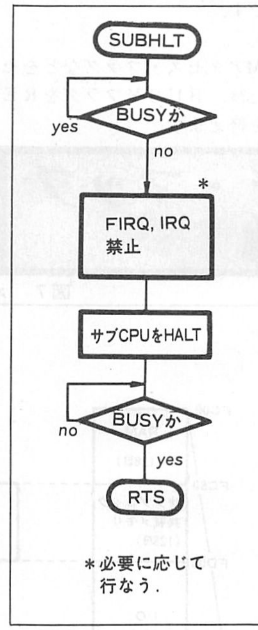


図4　サブCPU開始



図5　サブシステム出力

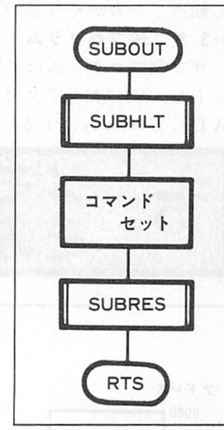
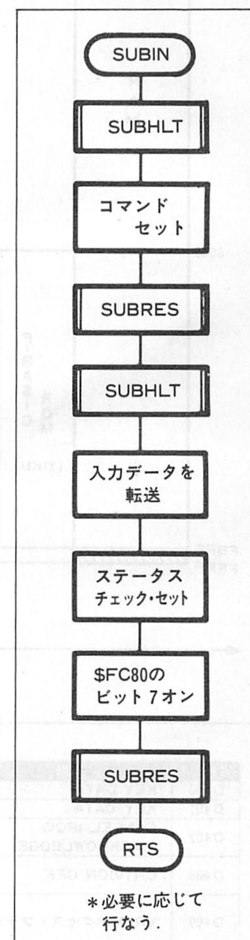


図6　サブシステム入力



4. メンテナンス・コマンド

サブシステムのモニタを使わず，新らたにプログラムを転送して，そのプログラムからVRAMを直接操作したい場合（高速ゲームなど）があります．このとき，メンテナンス・コマンドを使います．

4-1 キーワード

セットするキーワードは，＄59，＄41，＄4D，＄41，＄55，＄43，＄48，＄49で“YAMAUCHI”のASCII コードです．

4-2 プログラムの転送と実行

MOVEサブコマンドを用いて，プログラムをサブシステムに転送します．転送先がVRAM上のときは，VRAMセット・フラグをセットしてください．実行させるには，JMSRサブコマンドを使います．大きなプログラムを転送す

るには，最初に転送プログラムを送って，それを実行させて転送した方がやり易いようです．

4-3 サブ側プログラム

サブ側プログラムは，VRAMアクセス・フラグなどをセットし，VRAMをアクセスした後，ＢＵＳＹフラグをＲＥＡＤして(Readyにする) 処理を終えます．

5. メモリ・マップ その他

メモリ・マップを図７に示します．ブートＲＯＭ領域は，本体背面のディップ・スイッチ９，10により，４種類のうち１つを選択します．

メインとサブのＩ/Ｏアドレスを表１，２に示します．ここには，メインとサブ間のインターフェイスのフラグだけでなく，ライン・プリンタ，オーディオ・カセット，ブザー，ＲＳ-232Ｃ，ＡＤコンバータ，バブル・メモリ，ミニフロッピー，漢字ＲＯＭ（以上メイン・システム），キー入力データ，ブザー，上下スクローソング（以上サブシステム）などのフラグがあります．

図７　メモリ・マップ

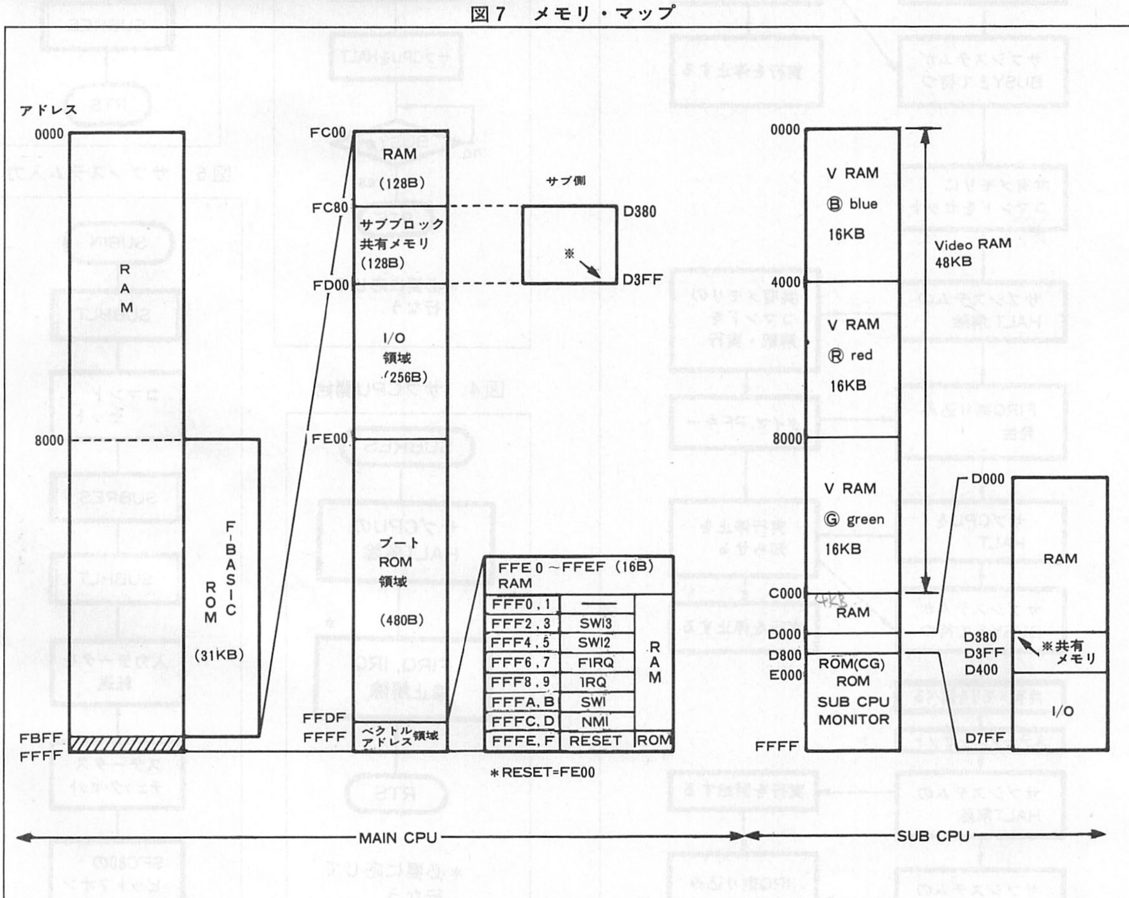


表２　サブＩ/Ｏアドレス

アドレス	用途	RESET	リード/ライト	D7	D6	D5	D4	D3	D2	D1	D0	信号
D400	KEY DATA		リード	KD8	＊3							正論理
D401	KEY DATA		リード	KD7	KD6	KD5	KD4	KD3	KD2	KD1	KD0	正論理
D402	CANCEL IRQの ACKNOWLEDGE		リード									‾_‾
D408	CRTのON/OFF	OFF (L)	リード＝ON									H
			ライト＝OFF									L
D409	VRAMアクセス・フラグ	RESET (L)	リード＝SET									H
			ライト＝ RESET									L
D40A	BUSYフラグ	BUSY (H)	リード＝ READY									L
			ライト＝ BUSY									H
D403	BUZZER		リード									‾_‾
D404	ATTENTION IRQ TO MAIN CPU		リード									‾_‾
D40D	INSERT MODE LED の ON/OFF	OFF (H)	リード＝ON									L
			ライト＝OFF									H
D40E	SCREEN RAMのOFFSET ADDRESS (H) ＊1	0 0	リード＊2 ＝禁止									
			ライト			OA13	OA12	OA11	OA10	OA9	OA8	

D40F	SCREEN RAMのOFFSET ADDRESS (L) ＊1	0 0	リード＊2 ＝禁止				
			ライト	OA7	OA6	OA5	

＊1　アドレスをセットするときは、VRAMアクセス・フラグをセットしておくこと。
＊2　READアクセスした場合、オフセット・レジスタの値は不定となる。
＊3　DATA欄が空欄の場合はDONT CARE

表1　メインI/Oアドレス

I/Oアドレス領域

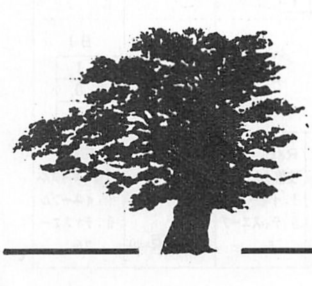
アドレス	内容	リード ライト	7	6	5	4	3	2	1	0
FD00	ライン・プリンタ(LP) オーディオ・カセット(O/C)	ライト	LP SLCTIN 0：セレクト状態 1：非セレクト	LP STRB ストローブ パルス	リザーブ				O/C リモート 0：リモートOFF 1：リモートON	O/2 MiC カセット ライト・データ
FD01	ライン・プリンタ・データ	ライト	D7	D6	D5	D4	D3	D2	D1	D0
FD02	オーディオ・カセット ライン・プリンタ	リード	O/C SPK カセット リード・データ		LP DET2 LPを識別するビット (未使用)	LP DET1	LP PE 1：用紙なし 0：正常	LP $\overline{\text{ACKNG}}$ 認知番号 (負論理)	LP $\overline{\text{ERROR}}$ 0：エラー 1：正常	LP BUSY 1：ビジー 0：レディー
FD03	ブザー	ライト	連続ブザー 0：OFF 1：ON	単一ブザー 0：OFF 1：ON (1Sec)						スピーカ 0：OFF 1：ON
FD04	サブインターフェイス	リード						PE 0：なし 1：パリティエラーあり	$\overline{\text{STOP}}$キー 0：ON 1：OFF	ATENT 1：ATTENT なし 0：〃あり
FD05	サブインターフェイス Z80	ライト	HALT 1：サブホールト イネーブル	CANCEL 1：サブIRQ リクエスト						Z80W 0：6809 1：Z80 が選択される
FD05		リード	BUSY 0：レディ 1：ビジィ							拡張検出 0：あり
FD06	RS-232C インターフェイス 受信データ	リード	D7	D6	D5	D4	D3	D2	D1	D0
FD06	RS-232C インターフェイス 送信データ	ライト	D7	D6	D5	D4	D3	D2	D1	D0
FD07	RS-232C インターフェイス ステータスレジスタ	リード	DSR 0：DSR信号H 1：DSR信号L	SYNDET 1：SYNC キャラクタ 検出	FE 1：フレーム エラー	OE 1：オーバーフローエラー	PE 1：パリティエラー	TXE 1：トランスミッタエンプティ	RXRDY 1：レシーバーレディ	TXRDY 1：トランスミットデータバッファエンプティ
FD07	RS-232C インターフェイス	ライト	モード・コマンド・データ(リセット後有効)							
FD07	RS-232C インターフェイス 同期モード	ライト	SCS SYNCキャラクタ 1：シングル 0：ダブル	ESD SYNC検出 1：外部 0：内部	EP パリティ 1：偶 0：奇	PE パリティ 1：イネーブル 0：ディスエーブル	L2 キャラクタ長 5: 0 0 6: 0 1 7: 1 0 8: 1 1	L1	0	0
FD07	RS-232C インターフェイス 非同期モード	ライト	S2	S1	同上	同上	同	上	B2 1/1: 0 1 1/16: 1 0 1/32: 1 1	B1
FD07	RS-232C インターフェイス コマンドデータ		EH ハントモード 1：SYNC キャラクタ サーチ	IR 内部リセット 1：リセット	RTS 1：RTS ＝0	ER エラーリセット 1：エラーフラグクリア	SBRK ブレーク検出 1：TXD ＝L	RXE 交信イネーブル 1：イネーブル 0：ディスエーブル	DTR ターミナル レディー 1：$\overline{\text{DTR}}$＝0	TXEN 送信イネーブル 1：イネーブル 0：ディスエーブル

I/O

表 1

アドレス	内　容	リード ライト	7	6	5	4	3	2	1	0
FD08		ライト			カウンタ・クリア 1：クリア		V RESET 1:DSPリセット	INT 1：有効	STOP 1：ストップ	START 1：スタート
FD09	ビット0〜3：データ ビット4〜7：コマンド	ライト	CM3	CM2	CM1	CM0	DATA3	DATA2	DATA1	DATA0
FD0A		リード	BUSY 0：BUSY 1：レディ							
FD0B	音声データ メモリ・バッファ	ライト	D7	D6	D5	D4	D3	D2	D1	D0
FD0C	A/Dコンバータ	リード								DOUT 変換データ
		ライト	ADC アナログ ブロック				$\overline{CS}$ 0：チップセレクト	RS レンジ・セレクト 1:0〜25V 0:0〜5V	C1 チャンネルセレクト A0 0 A1 0 A2 1 A3 1	C0 0 1 0 1
FD10	入出力データ	リード ライト	D7	D6	D5	D4	D3	D2	D1	D0
FD11	コマンド レジスタ	ライト	0	0	0	0	CM3	CM2	CM1	CM0
FD12	ステータス レジスタ	リード	CME 1:コマンド・エンド	TDRA 1：書込可	RDA 1：読出可				ERROR 1：エラー	BUSY 1：ビジー 0：レディ
FD13	エラーステータス レジスタ	リード	EJECT エラー		POUR ページ・オーズ	CRC エラー	トランスファ エラー	バッドループ オーバーエラー	ノーマーカ エラー	未定義 コマンド・エラー
FD14	バブル・メモリ	ライト	ページ・アドレス・レジスタ(H)							
FD15		ライト	〃　(L)							
FD16		ライト	ページ・カウント・レジスタ(H)							
FD17		ライト	〃　(L)							
FD18	ミニフロッピー	リード	ステータス・レジスタ							
FD19		ライト	コマンド・データ							
FD1A		リード ライト	トラック・レジスタ							
FD1B		リード ライト	セクタ・レジスタ							
FD1C		リード ライト	データ・レジスタ							
FD1D		リード ライト	ヘッド・レジスタ							
FD1E		リード	ドライブ・レジスタ							
FD1F		リード	DRQ フラグ	IRQ フラグ						
FD20	漢字ROM (JISコード)	ライト	漢字アドレス(H)							
FD21		ライト	漢字アドレス(L)							
FD22		リード	漢字データ(H)							
FD23		リード	漢字データ(L)							

＊FD24〜FDFFまで未使用



サブシステムに プログラムを転送する方法

■COMPAC O.H

FM-8の“インベーダー”と“ギャラクシアン”はいかがでしたか. I/O '82年5月号にそのKNOW-HOWの一部を発表しましたが，今回はサブCPUにプログラムを転送する方法およびそのプログラムを発表します.

サブシステム転送プログラム

このプログラム自体はサブルーチンの形式を取っているので，先頭番地（ここでは&H5000）をコールするとサブCPUにプログラムが転送され、サブCPUの制御は&HC000に移ります.

130行～170行でサブCPUをHLTし，割り込みを禁止しています．特に割り込みを禁止する必要はないと思いますが，とにかく入れてあります.

180行（実質は540～610行）で620行以降のメンテナンス・コマンドとそのデータおよびサブCPUへのプログラムを転送するプログラムを共通領域に書き込んでいます.

メンテナンス・コマンドのJMPサブコマンドにより，サブCPUは&HD38Fにジャンプします．この&HD38F以降（670～860行）では，VRAMアクセス・フラグをセットし，メインCPUからの転送データを待っています.

メインCPUは290行からサブCPUへプログラムを1バイトずつ転送しています．この辺のことはプログラムを見ていただいた方がよくわかると思います．また，実際のサブCPUへ転送するプログラムは“LABEL TABLE”の後へ書きます.

サブCPUのI/Oアドレス

&HD408 ……CRTのON/OFF
このアドレスをREADすると画面が表示され，WRITEすると画面が消える.
普通はREADした状態にしておく.

&HD409 ……VRAMアクセス・フラグ
このアドレスをREADするとセットされ，WRITEするとリセットされる.
普通はセットした状態（＝READ）にしておく.

&HD40A ……BUSYフラグ
このアドレスをREADするとメインCPUの&HFD05のビット7がリセット（READY）の状態になり，WRITEすると&HFD05のビット7がセット（BUSY）の状態になる.

終わりに

文章を書くのが苦手なので，かなり雑な説明となりましたが，説明を読んでいただくよりもプログラムを見ていただいた方がよく理解していただけると思います.

サブシステム転送プログラム

```
00010  5000                             ORG    $5000
00020                          *
00030                          *
00040                          *
00050               D386       SADD     EQU    $D386
00060               D381       TYPE     EQU    $D381
00070               D380       COMAND   EQU    $D380
00080               D382       MTYPE    EQU    $D382
00090               FD05       HLT      EQU    $FD05
00100                          *
00110                          *
00120                          *
00130  5000 B6      FD05       START    LDA    HLT
00140  5003 2B      FB    5000          BMI    START
00150  5005 1A      50                  ORCC   #$50
00160  5007 86      80                  LDA    #$80
00170  5009 B7      FD05                STA    HLT
00180  500C 8D      5D    506B          BSR    MOVE1
00190  500E 7F      FD05                CLR    HLT
00200  5011 8E      0000                LDX    #0
00210  5014 30      1F         L1       LEAX   -1,X
00220  5016 26      FC    5014          BNE    L1
00230  5018 B6      FD05       WAIT2    LDA    HLT
00240  501B 2B      FB    5018          BMI    WAIT2
00250  501D 33      8D 00A0             LEAU   PAT3,PCR
00260  5021 30      8D 00C9             LEAX   END,PCR
00270  5025 BF      2000                STX    $2000
00280  5028 C6      FF                  LDB    #$FF
00290  502A 86      80         SMOV     LDA    #$80
00300  502C B7      FD05                STA    HLT
00310  502F B7      FD05                STA    HLT
00320  5032 A6      C0                  LDA    ,U+
00330  5034 B7      FC81                STA    $FC81
00340  5037 F7      FC80                STB    $FC80
00350  503A 7F      FD05       SMOV2    CLR    HLT
00360  503D 7F      FD05                CLR    $FD05
00370  5040 B6      FD05       SMOV1    LDA    HLT
00380  5043 2B      FB    5040          BMI    SMOV1
00390  5045 86      80                  LDA    #$80
00400  5047 B7      FD05                STA    HLT
```

サブシステム転送プログラム

```
504A B7   FD05                 STA    HLT
504D B6   FC80                 LDA    $FC80
5050 26   E8   503A            BNE    SMOV2
5052 11B3 2000                 CMPU   $2000
5056 26   D2   502A            BNE    SMOV
5058 86   01                   LDA    #$1
505A B7   FC80                 STA    $FC80
505D 7F   FD05                 CLR    HLT
5060 7F   FD05                 CLR    HLT
5063 8E   0000                 LDX    #0
5066 30   1F         L3        LEAX   -1,X
5068 26   FC   5066            BNE    L3
506A 39                        RTS
506B 108E FC80       MOVE1     LDY    #$FC80
506F 30   8D 000A              LEAX   TABLE1,PCR
5073 C6   43                   LDB    #END1-TABLE1
5075 A6   80         MOV1      LDA    ,X+
5077 A7   A0                   STA    ,Y+
5079 5A                        DECB
507A 26   F9   5075            BNE    MOV1
507C 39                        RTS
507D      00         TABLE1    FCB    0,0,$3F
507E      00
507F      3F
5080      59                   FCC    /YAMAUCHI/
5081      41
5082      4D
5083      41
5084      55
5085      43
5086      48
5087      49
5088      93                   FCB    $93
5089      D38F                 FDB    PAT1-TABLE1+$D380
508B      90                   FCB    $90
          508C       PAT1      EQU    *
508C B7   D40A                 STA    $D40A
508F B6   D408                 LDA    $D408
5092 B6   D409                 LDA    $D409
5095 B6   D40A                 LDA    $D40A
5098 CE   C000       MOVE      LDU    #$C000
509B 7F   D380       MOVEA     CLR    $D380
509E 7F   D380                 CLR    $D380
50A1 7F   D380                 CLR    $D380
50A4 B6   D40A       MOV11     LDA    $D40A
50A7 B6   D380                 LDA    $D380
50AA 27   F8   50A4            BEQ    MOV11
50AC 81   01                   CMPA   #1
50AE 27   0D   50BD            BEQ    MOV21
50B0 B7   D40A                 STA    $D40A
50B3 B7   D40A                 STA    $D40A
50B6 B6   D381                 LDA    $D381
50B9 A7   C0                   STA    ,U+
50BB 20   DE   509B            BRA    MOVEA
50BD 7E   C000       MOV21     JMP    $C000
50C0      0001       END1      RMB    1
50C1 7F   D380       PAT3      CLR    $D380
50C4 10CE CFFF                 LDS    #$CFFF
50C8 7F   D380                 CLR    $D380
50CB 7F   D380                 CLR    $D380
50CE B6   D40A       PATA      LDA    $D40A
50D1 B6   D380                 LDA    $D380
50D4 27   F8   50CE            BEQ    PATA
50D6 B7   D40A                 STA    $D40A
50D9 B7   D40A                 STA    $D40A
50DC 30   8D 000C              LEAX   DATA-2,PCR
50E0 30   02         PA1       LEAX   2,X
50E2 A1   80                   CMPA   ,X+
50E4 26   FA   50E0            BNE    PA1
50E6 EC   84                   LDD    ,X
50E8 30   8D 0002              LEAX   DATA,PCR
50EC 6E   8B                   JMP    D,X
                     *
                     *
                     *
                     *
                     *
                     *
```

```
                    *
                    *
                    ********************************
                    *                              *
                    *     LABEL TABLE              *
                    *                              *
                    ********************************
          50EE      DATA    EQU    *
                    *FCB $E0
                    *FDB GMOVE-DATA
                    *FCB $50
                    *FDB BEAM-DATA
                    *FCB $20
                    *FDB KEYIN-DATA
                    *FCB $80
                    *FDB SPC8-DATA
                    *FCB $F1
                    *FDB BOM-DATA
                    *FCB $DD
                    *FDB BEAMM-DATA
                    *FCB $16
                    *FDB SPC16-DATA
                    *FCB $1
                    *FDB GB-DATA
                    *FCB $2
                    *FDB GM-DATA
                    *FCB $3
                    *FDB GR-DATA
                    *FCB $4
                    *FDB GK-DATA
                    *FCB $A0
                    *FDB STAR-DATA
                    *FCB $B0
                    *FDB MBO-DATA
                    *FCB $C0
                    *FDB MCO-DATA
                    *FCB $EE
                    *FDB YA-DATA
                    *FCB $33
                    *FDB PRMES-DATA
                    *FCB $34
                    *FDB MES-DATA
                    *FCB $35
                    *FDB CLEAR-DATA
                    *FCB $36
                    *FDB CLE-DATA
                    *FCB $39
                    *FDB MEN-DATA
                    *
                    *
                    *
                    *
                    *
                    *
                    *
                    *
                    *
                    *
                    *
                    *
                    *
                    *
                    *
                    *
                    *
                    *
                    *
                    *
                    *
50EE      0001      END     RMB    1
          5000              END    START
```

カセットサービス **FM-8**

Z80カードの使い方 Z80逆アセンブラ

■近藤孝吉
■川久保隆

長年使い慣れたMZ-80K（そう，あの便利なDISKとも）と別れて，ＦＭ-8と付合うようになり，ＭＺで作ったソフトを使わないのは，ＭＺがかわいそうと思い，移植したのがこのプログラムです．研究室にもＦＭ-8があるのと，68系に強い川久保氏の協力でパッチできましたので発表します．

なお，今回のパッチの際に使用した，6809アセンブラ，逆アセンブラおよびＭＺテープのＦＭ-8へのロード・プログラムは次の機会に発表します．乞うご期待！（すべてマシン語）．

Z80カード

図１を見るとわかると思いますが，RESET後は必ず6809から始まります．そしてFD05番地に01Ｈを書き込むと，6809のＰＣはその命令の次の番地を示したままHALTになり，Ｚ80に実行が移ります．Ｚ80はRESETがかかったままWAITになっていたので，ＰＣは０番地であり，ここから実行を始めます．

その後は図に示す通りで，切り換えデータをＦＤ05番地に書き込む命令を実行した後のＰＣを示したまま，移行するのです．

注意していただきたいのは，移行したときのＰＣを考えながらプログラムを作らなければならないことです．図２にそのプログラム例を示します．

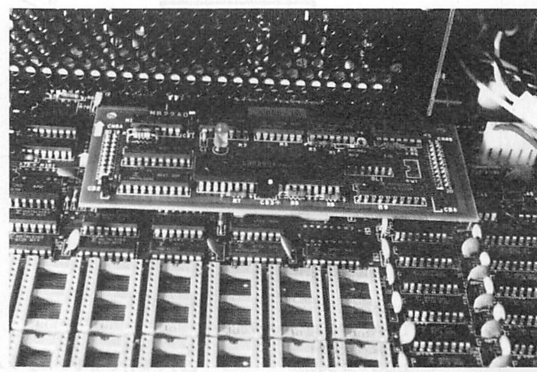
まず，6809で０番地からＺ80用のＪＵＭＰ命令を書いておきます．これはRESET後のために，そしてＺ80から6809に戻る命令の次に，再度，Ｚ80に移行したときに，START番地へ飛ぶ命令を書いておきます．こうしておけば，常にＺ80へ移行したとき決った番地より実行することになります．

ＦＭ-8ではＺ80使用時のことを考えてか，０番地からと38(16)番地からと66(16)番地からそれぞれ３バイト分，使っていないようなので，RST 38Hや，NMI，などが使えるようです．

なお，研究室のＦＭ-8ではＺ80から移行してきたときに実行しない（1バイトをフェッチし忘れる？）命令があるので，ＮＯＰを入れるようにしています．皆さんのＦＭ-8ではいかがでしょうか．一度調べてみては………．

Z80ディスアセンブラ

プログラムは，参考文献１）に改良を加えたものに，さらにＦＭ-8用にパッチし，ＢＩＯＳを通すように6809のプログラムと合わせてあります．



《使用法》

リストを打ち込んでから走らせる前にセーブしておいてください．一度走らせると自分自身でプログラムを書き変えているため，セーブしても次のロード後は正常に動かなくなるので，

SAVEM "Z80DISAS", &H5000, &H5DAF, &H5000

としてセーブしましょう．

コマンドは表1に示すとおりで，特に説明はいらないと思います．なお，一度走らせると，**PF･5** が"EXEC" ＋CHR$(13)となるようになっているので，その後BASICに戻って，再度走らせる場合は（EXECアドレスを変えていなければ）**PF･5** を押せばよいのです．

それと自動的にWIDTH40，25　CONSOLE０，25，０，０　COLOR　７を実行するので，80字にしたい方は，セーブ前に5D6F番地を50Ｈにしてください．カラーは5D89番地に表示したい色のカラーコード（01Ｈ～07Ｈ）を書いてください．

図３はＣＲＴへの出力のハード・コピー，図４はPRINTERへ出力したものです．ＣＲＴへの出力は家庭用テレビ使用時を考えて40字モードにしていますが，詰まっていて見にくいという方は上記のようにしてください．

なお，PRINTERへの出力時ページングは，フォーム・フィードを使わず，ＭＰ-80のステップ・パーフォレーションを使っています．これは，ロール紙を使う場合に，少しでも紙の使用量を減らすためです．

また，Ｚ80のＳＰは６Ｅ00Ｈに，ラベル・エリアはプロ

図1　6809，Z80切り換え

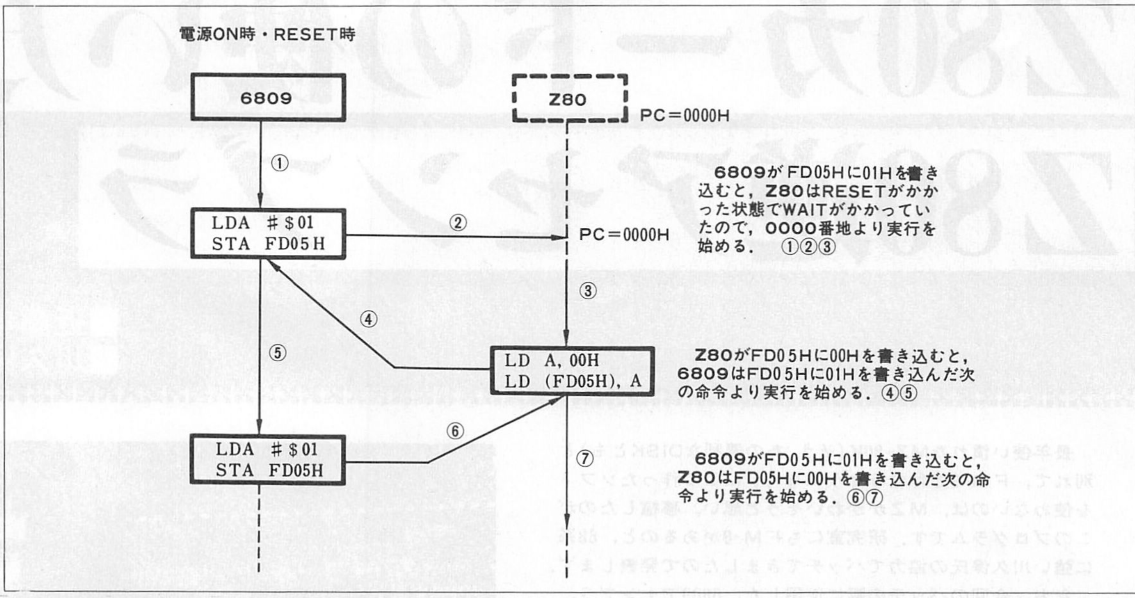


図2　プログラム例

```
EXEC & H6000の場合
6809
     6809                   Z80
6000 LDA  ＃$C3       7000
     STA  0000H
     LDD  ＃$0070            Z80で実行させるプログラム
     STD  0001H
     LDA  ＃$01              LD   A, 00H
     STA  FD05H              LD   (FD05H), A
     RTS                     JP   7000H
```

図3

```
  **  Z80 DISASSEMBLER  **
 ]C
  CRT MODE
 ]A
  FROM 5100  TO 5200
 ]R
  REGISTER LABEL                              アスキー
  END             ラベル                      ダンプ
 ]D
 5100 ED7B0E51 L5100:LD   SP,(L510E)o( Q
 5104 C3C151          JP   L51C1      テチQ
 ;
 5107 01211B          LD   BC,L1B21    !
 510A AD       L510A:XOR  L           ュ
 510B 5B              LD   E,E        [
 510C 00              NOP
 ]!
 Ready
 HARDC
```

グラム・エンドから6D00Hまで，そしてプログラムはスタート後5000H～6FFFHまでに広がって入っているので，誤ってつぶさないように！

EPSON MP-80について

MP-80シリーズは，TYPE1，2ともに，メカは同じで内部のソフトが違うだけです．また，TYPE2のPC用は普通のTYPE2のディップ・スイッチの設定を変えてケーブルをつけているだけです．

表1　コマンド

A	番地指定
C	CRTモードにする．
D	ディス・アセンブル実行．
N	ラベル・エリア・クリア
P	PRINTERモードにする．
R	ラベル登録
SPACE	ディス・アセンブル中の一時停止．再度押すと続行．
CTRL-X	ディス・アセンブル中の中断，BREAK，コマンド待ちへ．
CTRL-C STOP	ディス・アセンブル中，およびコマンド待ちよりBASICへリターン．
!	コマンド待ちのとき，BASICへリターン．
CTRL-R	出力をCRTとPRINTERの両方へ行なうモードにする．
CTRL-Q	上記を解除する．

BASICからのスタートはEXEC ＆H5000です．

図4

```
  **  Z80 DISASSEMBLER  PAGE 01  **

 5100 ED7B0E51    L5100: LD    SP,(L510E)    o( Q
 5104 C3C151             JP    L51C1         テチQ
                  ;
 5107 01211B             LD    BC,L1B21       !
 510A AD          L510A: XOR   L              ュ
 510B 5B                 LD    E,E            [
 510C 00                 NOP
 510D 6D                 LD    L,L            m
```

これはMB27401についても同じで，実は普通のTYPE2なのです．だから，FM-8用にプリンタを買われる方はMP-80F/T TYPE2とケーブルMB26504を買ってディップ・スイッチの1-5，1-7，2-1，2-2すべてを，OFFにすればよいのです．普通紙も使えて便利です．

ただし，MPにもいろいろバージョンがあるらしく，初期のものはESC M(96桁モード）があるのに，現在のものはないので，確実とは言いきれませんが，私の知っている3台はOKでした．

おわりに

6809-Z80間は，どんなプログラムからでも移行ができるようにしてあり，ほかのプログラムからでも（Z80から）BIOSをコールすることができます．このプログラムにつ

いては，次の機会に発表します．

以上いろいろと書きましたが，ＦＭ-8をお使いの方の参考になれば幸いです．

参考文献

1）インターフェース，80年４月，p.131，CQ出版社

2）ＦＭ-8ユーザーズ・マニュアル・システム解説，富士通．

Z80逆アセンブラ　ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5000 7E 50 15 12 12 12 12 12 12 12 12 12 12 12 12 12 :CD
5010 12 12 12 12 00 B6 50 14 26 0F 8E 5B AE CE 6E 00 :6A
5020 A6 80 A7 C0 8C 5D AE 26 F7 86 01 B7 50 14 7E 50 :B1
5030 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :40
5040 7E 6E 9F 8E 50 75 63 80 8C 50 8E 26 F9 8E 50 6D :95
5050 AD 9F FB FA 8E 50 75 63 80 8C 50 8E 26 F9 8E 00 :8E
5060 00 86 09 4A 26 FD 30 01 26 F7 7E 6E 6A 10 00 50 :00
5070 75 00 20 00 80 F5 FF E6 F9 FF FF F9 FB FF FF FF :D7
5080 CD F3 B4 BE B1 AC BE B6 DF AA B1 B6 A9 D1 00 00 :6D
5090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
50A0 4B 41 4E 53 41 49 20 55 4E 49 56 2E 0D 0A 86 20 :04
50B0 B7 03 13 AD 9F FB FE 12 6F 00 8E 20 20 AD 9F FB :A8
50C0 FC 54 4A 20 FE 16 FF FD 7E 50 C8 00 00 00 00 00 :60
50D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
50E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
50F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
---------------------------------------------------------
Sum: E1 00 F0 94 B1 E2 F2 30 74 BC 59 43 6A 12 00 39 :9B


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5100 ED 7B 0E 51 C3 C1 51 01 21 1B AD 5B 00 6D 00 6E :BC
5110 02 5B FF 89 94 C9 51 AF 32 05 FD 18 E3 FF 96 6E :74
5120 C9 E1 FF 06 73 DD E1 C9 32 09 51 AF 32 05 FD F1 :09
5130 C9 D5 57 FF 90 6E D1 C9 D5 16 07 05 FD 3A 08 51 :13
5140 C9 D5 16 07 C3 33 51 F1 C9 FD C9 D5 FF 93 6E 7A :D1
5150 D1 C9 33 2F 51 F1 C9 C9 D5 FF B2 6E D1 C9 16 0D :81
5160 18 D1 D5 FF 9C 6E 7A D1 FE 01 C9 3A 08 51 FE 01 :6C
5170 C9 B1 D8 B6 DE 20 C4 B3 20 BA DE 20 BB DE 20 B2 :C0
5180 CF BC C0 20 CF C0 C9 20 B5 BA BC A6 20 B5 CF C1 :19
5190 BC C3 B2 20 CF BD 20 20 B6 DD BB B2 C0 DE 20 B2 :8D
51A0 B6 DE 20 B8 20 BE B2 B7 DE AE 20 B6 B2 DB 20 B9 :7B
51B0 DD B7 AD B3 BC C2 4B 41 4E 53 41 49 20 55 4E 49 :35
51C0 56 11 1B 53 CD 1D 51 21 AD 5B 22 0A 51 CD C6 54 :9D
51D0 21 00 00 22 49 5B 22 4B 5B AF 32 55 5B ED 7B 0E :B6
51E0 51 CD 58 51 3E 5D CD 31 51 CD 41 51 CD 4B 51 CD :46
51F0 31 51 FE 52 28 3B FE 44 28 3C FE 50 28 1D FE 43 :AF
---------------------------------------------------------
Sum: 13 EF 09 8D DE 94 D0 99 2E A1 8F 1B F8 1B 2A 3F :68


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5200 28 23 FE 21 CA 12 51 FE 41 CA 3B 52 FE 4E 20 CD :66
5210 CD 7E 53 11 37 53 CD 4B 54 18 C2 11 3C 53 CD 4B :37
5220 54 3E 01 18 07 11 4A 53 CD 4B 54 AF 32 55 5B 18 :75
5230 AC CD 54 53 18 A7 CD B1 53 18 A2 CD 58 51 11 6E :5F
5240 54 CD 1D 51 CD 74 52 38 22 22 49 5B 22 4D 5B 11 :1D
5250 75 54 CD 1D 51 CD 74 52 38 11 22 4B 5B CD 31 54 :FA
5260 D2 DD 51 CD 41 51 11 AC 53 18 03 11 A5 53 CD 4B :AB
5270 54 C3 DD 51 06 04 11 5C 5B CD 4B 51 FE 03 28 1D :C6
5280 4F CD 9F 52 38 10 79 CD 31 51 79 12 13 10 EA 11 :C6
5290 5C 5B CD B6 52 C9 D5 CD 41 51 D1 18 DC 37 C9 C5 :13
52A0 E5 01 00 10 21 0B 53 BE 20 03 79 18 06 23 0C 05 :21
52B0 20 F5 37 E1 C1 C9 D5 CD C5 52 38 07 67 CD C5 52 :FA
52C0 38 01 6F D1 C9 C5 1A 13 CD 9F 52 38 0D 07 07 07 :4C
52D0 07 4F 1A 13 CD 9F 52 38 01 B1 C1 C9 D5 E5 21 0B :9B
52E0 53 E6 0F 5F 16 00 19 7E E1 D1 C9 7C CD F4 52 7D :DB
52F0 CD F4 52 C9 F5 E6 F0 0F 0F 0F 0F CD DC 52 CD 31 :DC
---------------------------------------------------------
Sum: F3 B5 4B 2E 92 AA 08 DC D2 84 92 7A CB 20 A5 58 :8B


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5300 51 F1 E6 0F CD DC 52 CD 31 51 C9 30 31 32 33 34 :44
5310 35 36 37 38 39 41 42 43 44 45 46 0C 20 20 2A 2A :48
5320 20 20 5A 38 30 20 44 49 53 41 53 53 45 4D 42 4C :09
5330 45 52 20 20 2A 2A 0D 20 4E 45 57 0D 20 50 52 49 :5A
5340 4E 54 45 52 20 4D 4F 44 45 0D 20 43 52 54 20 4D :01
5350 4F 44 45 0D 2A 49 5B 22 4D 5B 11 84 53 CD 4B 54 :D1
5360 CD 62 51 30 19 CD CC 54 CD AA 54 38 0B CD 31 54 :16
5370 30 EE 11 94 53 C3 4B 54 11 99 53 C3 4B 54 2A 0A :0B
5380 51 C3 C6 54 20 52 45 47 49 53 54 45 52 20 4C 41 :60
5390 42 45 4C 0D 20 45 4E 44 0D 20 4C 41 42 45 4C 20 :84
53A0 4F 56 45 52 0D 20 42 52 45 41 4B 0D 20 3F 3F 3F :B8
53B0 0D 2A 49 5B 22 4D 5B 3A 55 5B FE 01 20 37 11 55 :4B
53C0 54 CD 1D 51 CD 58 51 11 6E 54 CD 1D 51 2A 49 5B :E1
53D0 CD EB 52 11 75 54 CD 1D 51 2A 4B 5B CD EB 52 CD :C6
53E0 58 51 21 01 00 22 5A 5B CD A1 5A 3A 57 5B FE 01 :55
53F0 C8 AF 32 56 5B CD 62 51 D0 CD CC 54 CD 3C 54 3A :2E
---------------------------------------------------------
Sum: B5 C1 E5 89 22 2C B0 78 D2 C2 B8 F8 C7 B8 8C 4A :F3


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5400 57 5B FE 01 C8 DD CB 00 4E 28 09 CD 1A 54 3A 57 :6C
5410 5B FE 01 C8 CD 31 54 30 DC C9 CD 7B 54 21 3B 0D :4E
5420 3A 55 5B FE 01 28 05 22 6A 5B 18 10 22 6E 5B 18 :28
5430 0B 2A 4B 5B ED 5B 4D 5B B7 ED 52 C9 AF 32 57 5B :1D
5440 3A 55 5B FE 01 11 5C 5B CA 1B 5A CD 58 51 CD 1D :50
5450 51 00 00 00 C9 20 44 49 53 41 53 53 45 4D 42 4C :21
5460 49 4E 47 20 28 50 52 49 4E 54 45 52 29 0D 20 46 :E6
5470 52 4F 4D 20 0D 20 20 54 4F 20 0D 21 5C 5B 06 50 :59
5480 3E 20 77 23 10 FC FD 21 5C 5B C9 ED 5B 0A 51 1A :5F
5490 FE FF 28 0D BD 13 13 20 F6 1B 1A 13 BC 20 F0 37 :76
54A0 C9 13 1A FE FF 1B 1A 20 EB C9 B7 DD CB 00 46 C8 :69
54B0 2A 52 5B CD 8B 54 3F D0 E5 2A 0C 51 B7 ED 52 E1 :D5
54C0 D8 EB 73 23 72 23 36 FF 23 36 FF C9 CD 7B 54 CD :AD
54D0 D5 55 CD 8B 54 30 1C FD 21 6A 5B 3A 55 5B FE 01 :EE
54E0 20 04 FD 21 6E 5B 3E 4C CD 90 57 CD D5 55 3E 3A :B8
54F0 CD 90 57 AF 32 51 5B 32 54 5B DD 21 54 5B 2A 4D :46
---------------------------------------------------------
Sum: E6 22 3C D9 3F AF D7 99 8C FD 73 D3 45 B8 EF 25 :5B


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5500 5B 22 4F 5B CD 1E 56 FE CB 28 11 FE ED 28 15 FE :90
5510 DD 28 19 FE FD 28 19 CD 5B 56 18 2D CD 1E 56 21 :7F
5520 21 59 18 29 CD 1E 56 21 5D 59 18 21 3E 16 18 02 :7A
5530 3E 18 32 51 5B CD 1E 56 FE CB 28 58 01 27 00 21 :07
5540 EF 59 ED B1 20 2E CD 69 56 4F 21 17 58 CD 2B 56 :ED
5550 38 22 CD E0 55 FD 21 61 5B ED 5B 4D 5B 2A 4F 5B :FA
5560 22 4D 5B 06 00 04 CD 1E 56 CD 81 57 3A 4D 5B BB :57
5570 20 F3 18 40 21 13 58 CD 68 57 2A 4F 5B 22 4D 5B :21
5580 FD 21 75 5B 3A 55 5B FE 01 20 04 FD 21 7B 5B CD :BC
5590 6F 56 18 C1 CD 1E 56 CD 1E 56 21 21 59 CD 2B 56 :09
55A0 38 D2 ED 5B 4F 5B 13 13 ED 53 4D 5B CD E0 55 CD :D9
55B0 1E 56 18 A1 3A 55 5B FE 01 FD 21 7F 5B 20 04 FD :2F
55C0 21 8A 5B 7E FE 20 30 02 3E 20 CD 90 57 23 10 F3 :0C
55D0 3E 0D C3 90 57 2A 4D 5B 7C CD 81 57 7D C3 81 57 :00
55E0 5F CD 68 57 7B E6 3F C8 16 00 FD 21 75 5B 3A 55 :E6
55F0 5B FE 01 20 04 FD 21 7B 5B 21 96 57 19 19 5E 23 :33
---------------------------------------------------------
Sum: DB 77 F8 47 EC C3 F2 73 28 D6 04 05 45 8B AD B8 :E1


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5600 7E F5 7B CD 0F 56 F1 FE 6E C8 16 2C CD 7B 57 21 :47
5610 6E 56 BD 6F 30 01 24 79 1F 1F 1F E6 07 E9 D9 2A :F4
5620 4D 5B 23 22 4D 5B 2B 7E D9 4F C9 AF BE 28 0D 79 :4A
5630 BE 23 3E 00 C8 CB 7E 23 28 FB 18 EF 23 7E 23 A1 :E2
5640 BE 23 7E 23 28 0D CB 7E 20 F3 7E 87 23 38 EE 20 :81
5650 F9 37 C9 CB 7E C8 23 23 23 18 F8 21 16 5A 01 05 :1A
5660 00 ED B1 C0 DD CB 00 CE C9 FE E9 C0 18 F6 C9 CD :E8
5670 1E 56 CD 81 57 3E 48 18 4F CD 1E 56 41 CD 1E 56 :C9
5680 16 4C CD 7B 57 32 53 5B CD 81 57 78 32 52 5B CD :AA
5690 81 57 DD CB 00 C6 C9 79 E6 38 18 D6 3E 49 CB 59 :3F
56A0 28 02 3E 52 18 22 C3 8D 57 3E 30 11 3E 31 11 3E :D8
56B0 32 18 15 CD 79 57 CD CB 56 18 3F CD 79 57 CD E2 :8D
56C0 56 18 37 CD CE 56 3E 27 C3 90 57 3E 0E 11 3E 20 :60
56D0 11 3E 26 11 3E 0A 11 3E 0C 11 3E 07 18 75 FE 06 :10
56E0 28 EC E6 3E C6 08 FE 0C 20 69 3A 51 5B C6 0C 18 :69
56F0 62 CD 79 57 CD 1E 56 CD 72 56 3E 29 18 CA CD 79 :64
---------------------------------------------------------
Sum: AE 32 17 65 B5 52 43 09 AA 76 7E 59 07 98 4F AA :3E


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5700 57 C5 CD 79 56 C1 18 F2 CD 79 57 CD EA 56 18 EA :2F
5710 E6 03 87 C6 10 18 3C CD 1E 56 3A 4D 5B 81 47 3A :BF
5720 4E 5B CE 00 CB 79 28 01 3D C3 80 56 79 E6 07 FE :1E
5730 06 20 20 CD 79 57 CD EA 56 3A 51 5B B7 28 BB C5 :35
5740 CD 1E 56 C1 B7 16 2B F2 4E 57 16 2D ED 44 CD 7B :4D
5750 57 18 A4 21 EA 57 85 6F 30 01 24 7E E6 7F FE 20 :BF
5760 C4 90 57 BE F8 23 18 F3 FD 21 70 5B 3A 55 5B FE :60
5770 01 20 E8 FD 21 75 5B 18 E2 16 28 FD 72 00 FD 23 :BE
5780 C9 F5 0F 0F 0F 0F CD DC 52 CD 90 57 F1 CD DC 52 :95
5790 FD 77 00 FD 23 C9 6E 6E 2C 6E 6F 6E DA 6F E2 6E :49
57A0 A6 2C 2F 6E D7 E2 EA E2 2F 2C 2F 6F E2 79 EA FE :30
57B0 FE EA E2 FE FE E2 CB EA BB DA DA BB 12 79 79 6E :F9
57C0 08 6E 10 17 17 6E D4 D7 B3 EA CE C3 97 6E DA F1 :CB
57D0 2F D1 F1 DA D1 2F DE 6E DA 9C 9C DA DA FE FE DA :B3
57E0 12 6E DA 2C A9 6E AC 6E AF 6E C2 C3 C4 C5 C8 CC :76
57F0 CD C1 42 C3 44 C5 48 CC 53 D0 4E DA 5A A0 4E C3 :06
---------------------------------------------------------
Sum: FA 19 B8 01 40 1A 02 AB D2 60 B6 F7 42 FC 53 29 :6C


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5800 43 A0 50 CF 50 C5 50 A0 4D A0 41 C6 49 D8 49 D9 :3E
5810 28 43 A9 44 45 46 C2 3F 43 43 C6 2F 43 50 CC 27 :E5
5820 44 41 C1 F3 44 C9 FB 45 C9 D9 45 58 D8 76 48 41 :9C
5830 4C D4 17 52 4C C1 07 52 4C 43 C1 1F 52 52 C1 0F :D2
5840 52 52 43 C1 37 53 43 C6 00 F8 88 26 FF CE 03 41 :F2
5850 44 C3 F8 80 26 FF C6 03 CF 09 08 41 44 C4 F8 A0 :2E
5860 01 FF E6 02 41 4E C4 C7 C4 13 FF CD 14 43 41 4C :89
5870 CC F8 B8 01 FF FE 02 43 D0 C7 05 06 CF 0B 04 44 :83
5880 45 C3 FF 10 17 44 4A 4E DA FF EB 18 FF E3 19 FF :E0
5890 08 1A 45 D8 FF DB 1C 49 CE C7 04 06 CF 03 04 49 :3C
58A0 4E C3 FF E9 15 C7 C2 13 FF C3 14 4A D0 E7 20 16 :B7
58B0 FF 18 17 4A D2 EF 02 11 EF 0A 12 C0 40 09 C7 06 :2D
58C0 0A CF 01 0B FF 2A 0C FF 22 0D FF 3A 23 FF 32 24 :F9
58D0 FF F9 10 4C C4 FF D3 1E 4F 55 D4 FF 00 00 4E 4F :1C
58E0 D0 F8 B0 01 FF F6 02 4F D2 CF C1 20 50 4F D0 CF :7F
58F0 C5 20 50 55 53 C8 FF C9 00 C7 C0 25 52 45 D4 C7 :4B
---------------------------------------------------------
Sum: 96 9C 15 64 D4 EF ED 39 E1 65 0A 4C 7F 39 86 2E :9C
```

Z80逆アセンブラ　ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5900 C7 1B 52 53 D4 F8 98 26 FF DE 03 53 42 C3 F8 90 :D1
5910 01 FF D6 02 53 55 C2 F8 A8 01 FF EE 02 58 4F D2 :4B
5920 00 00 C0 40 05 42 49 D4 C0 80 05 52 45 D3 C0 C0 :93
5930 05 53 45 D4 F8 10 01 52 CC F8 18 01 52 D2 F8 00 :C5
5940 01 52 4C C3 F8 08 01 52 52 C3 F8 20 01 53 4C C1 :43
5950 F8 28 01 53 52 C1 F8 38 01 53 52 CC 00 A9 43 50 :65
5960 C4 B9 43 50 44 D2 A1 43 50 C9 B1 43 50 49 D2 AA :2C
5970 49 4E C4 BA 49 4E 44 D2 A2 49 4E C9 B2 49 4E 49 :56
5980 D2 A8 4C 44 C4 B8 4C 44 44 D2 A0 4C 44 C9 B0 4C :21
5990 44 49 D2 44 4E 45 C7 BB 4F 54 44 D2 B3 4F 54 49 :10
59A0 D2 AB 4F 55 54 C4 A3 4F 55 54 C9 4D 52 45 54 C9 :9E
59B0 45 52 45 54 CE 6F 52 4C C4 67 52 52 C4 00 CF 4A :B7
59C0 07 41 44 C3 FF 46 27 FF 56 28 FF 5E 29 49 CD C7 :9B
59D0 40 1D 49 CE CF 4B 0E CF 43 0F F7 57 21 F7 47 22 :8C
59E0 4C C4 C7 41 1F 4F 55 D4 CF 42 07 53 42 C3 00 86 :A5
59F0 09 8E 19 96 21 9E 22 A6 23 AE 29 B6 34 BE 35 2B :CF
---------------------------------------------------------
Sum: 9C 8C A0 22 3D 36 36 C5 AF 87 8D 07 AB 6C 1E 68 :BF


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5A00 36 E1 39 E3 46 E5 4E E9 56 F9 5E 6E 66 71 70 73 :6A
5A10 72 75 74 7E 77 2A C3 18 C9 76 E9 CD 39 5A 3A 57 :6E
5A20 5B FE 01 C8 21 56 5B 34 7E FE 36 D8 36 00 2A 5A :6C
5A30 5B 2C 22 5A 5B CD A1 5A C9 ED 73 58 5B F5 D5 1A :E6
5A40 CD 53 5A 1A FE 0D 28 03 13 18 F4 3E 0A CD 53 5A :AB
5A50 D1 F1 C9 F5 AF 32 57 5B CD 6A 5A F1 32 01 FD 3E :03
5A60 00 32 00 FD 3E 40 32 00 FD C9 D9 57 01 00 00 3A :10
5A70 02 FD E6 01 BA 20 02 D9 C9 1E 14 1D 20 FD 0B 78 :53
5A80 B1 20 EC D9 CD 58 51 11 25 5B CD 1D 51 3E 01 32 :49
5A90 57 5B ED 7B 58 5B C9 ED 73 58 5B F5 CD 53 5A F1 :09
5AA0 C9 CD B7 5A 06 50 21 D5 5A 7E CD 97 5A 3A 57 5B :75
5AB0 FE 01 C8 23 10 F3 C9 E5 2A 5A 5B 7D FE 0A 38 03 :3A
5AC0 24 AF 6F CD DC 52 32 F9 5A 7C CD DC 52 32 F8 5A :BD
5AD0 22 5A 5B E1 C9 0A 0A 0A 20 20 20 20 20 2A 2A 20 :B3
5AE0 20 5A 38 30 20 44 49 53 41 53 53 45 4D 42 4C 45 :2E
5AF0 52 20 20 50 41 47 45 20 30 31 20 20 2A 2A 20 20 :04
---------------------------------------------------------
Sum: 85 BF 53 8F 1F AE 8E F4 13 6E DB 95 EC 28 7C E8 :DE


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5B00 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 :00
5B10 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 :00
5B20 20 20 0A 0A 0A 20 4E 4F 20 50 4F 57 45 52 20 4F :37
5B30 52 20 4E 4F 20 43 4F 4E 4E 45 43 54 49 4F 4E 28 :47
5B40 50 52 49 4E 54 45 52 29 0D 00 00 00 00 CB 72 C8 :5F
5B50 72 00 22 72 03 00 05 00 FA 7C 01 00 20 20 20 20 :05
5B60 20 20 20 20 20 20 20 20 20 20 20 20 20 20 3B 0D :08
5B70 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 :00
5B80 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 :00
5B90 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 :00
5BA0 20 20 20 20 20 20 20 20 20 20 20 20 00 FF D9 E1 :39
5BB0 46 23 4E 23 ED 43 06 00 E5 D9 F5 E5 3E 01 21 05 :0D
5BC0 00 73 2B 72 2B 77 AF 32 05 FD 00 7E 23 56 23 5E :0D
5BD0 87 28 15 C5 DD E5 FD E5 23 46 23 6E 60 CD 08 00 :5C
5BE0 AF FD E1 DD E1 C1 18 D6 E1 F1 C9 31 00 6E 21 03 :58
```

```
5BF0 00 18 D8 34 6F EE 68 EC C1 EF 68 DD 06 86 01 9F :F6
---------------------------------------------------------
Sum: 90 45 EA 64 A6 D6 06 7F 04 ED BC 6A 35 63 22 F2 :E7


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5C00 04 97 03 86 01 B7 FD 05 12 9E 04 96 03 27 07 AD :06
5C10 9F 00 06 4F 20 E9 35 EF 86 C3 97 00 97 38 30 8C :8C
5C20 CA 1F 10 1E 89 DD 01 30 88 C3 1F 10 1E 89 DD 39 :E5
5C30 86 E9 97 08 8E 72 F0 8D BA 51 00 39 F2 D8 16 00 :AF
5C40 35 16 00 41 16 00 56 16 00 53 16 00 43 16 01 32 :03
5C50 16 01 2F 16 01 2C 16 00 03 7E 50 43 8E 01 00 39 :7B
5C60 34 10 30 8C 34 CC 0A 0D ED 84 C6 04 E7 02 8D 02 :CA
5C70 35 90 34 7F 20 2B 34 10 1F 10 30 8C 1C C6 0D ED :CE
5C80 84 8D 1A 35 90 86 01 A7 E2 17 00 63 1F 01 35 84 :53
5C90 6F E2 8D 5B 8E 00 00 35 84 8D 00 9F 86 34 7F C6 :AB
5CA0 0D 32 7E AF E4 CE 00 00 E1 80 27 04 33 41 20 F8 :36
5CB0 30 8D 00 9F 86 14 A7 84 6F 01 CC 00 02 ED 04 31 :81
5CC0 8D 00 B3 10 AF 02 B6 01 E5 A7 21 AD 9F FB FA 6F :15
5CD0 01 EF 04 27 16 EE E4 EF 02 AD 9F FB FA 6D 8D 00 :2F
5CE0 71 27 08 86 0E A7 84 AD 9F FB FA 32 62 35 FF 30 :98
5CF0 8D 00 68 33 8D 00 6C 86 15 A7 84 6F 01 EF 02 AD :F5
---------------------------------------------------------
Sum: 63 9A 8F 2B 8B 11 FF 67 3A F5 47 01 54 8E 25 8B :C2


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5D00 9F FB FA 7D 03 13 27 06 4F 1F 8B 7E D8 C2 6D 62 :34
5D10 27 06 6D 8D 00 4E 27 D7 A6 8D 00 47 81 1B 27 CF :7F
5D20 81 03 26 06 4F 1F 8B 7E D8 CA 81 11 26 03 6F 8C :7F
5D30 21 81 12 26 03 A7 8C 1A 81 18 26 05 32 63 16 FF :98
5D40 15 6D 62 2E 00 2B 08 81 20 26 04 63 62 20 A0 6F :04
5D50 62 39 00 14 00 00 00 00 00 00 00 15 00 00 00 00 :C4
5D60 00 00 00 00 00 00 00 00 00 00 00 0A 00 01 00 28 :33
5D70 19 00 19 00 01 00 11 04 0A 00 2A 05 05 45 58 45 :68
5D80 43 0D 30 8C CE 33 8C F0 86 07 B7 01 E5 EF 02 86 :2A
5D90 10 A7 84 6F 01 86 80 A7 05 AD 9F FB FA 33 8C CB :28
5DA0 EF 02 6F 01 AD 9F FB FA 6F 8C A7 16 FE AB 00 00 :03
5DB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5DC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5DD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5DE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5DF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
---------------------------------------------------------
Sum: 3A E1 3D 74 D2 AA 85 8B 72 F4 5D 74 F5 76 9F E9 :82
```

FM-8

6809 Z80 COMMUNICATER

汎用入出力ルーチン

■川久保　隆
■中井康博

FM-8を持っている人で，Ｚ80ソフトカードを持っている人は多いと思います．しかし，Ｚ80ソフトカードの使用にあたって，マニュアルにＺ80の起動法が書かれていないため，使えないでいる人も多いのではないでしょうか．

Ｚ80を起動するのには，FD05番地に1を書き込み，6809に戻すには，FD05番地に0を書き込むだけでよい（注1)のですが，それだけでは，6809でＺ80のPCを制御できないため，工夫しないと『リセットをかけなければ動作しないプログラム』などしかできなくて，非常に不便です．

はじめに

6809↔Ｚ80 COMMUNICATERは，FM-8上で6809からＺ80のサブルーチンをコールしたり，またその逆のコールをするためのプログラムで，これはリセット直後でも，他のプログラム（Ｚ80の）を実行した後でも，同様に働きます．また相手側プロセッサのサブルーチン・エントリ・アドレスを用意するだけで動作させることができて，あたかもサブルーチンをコールするだけのように扱えるプログラムです（注2）．

また合わせて，我々が各種プログラム（6809用アセンブラ，ディスアセンブラ，トレーサ，Ｚ80用アセンブラ，ディスアセンブラなど）で使っている入出力ルーチンも，紹介します．もちろん，この入出力ルーチンも，6809-Ｚ80 COMMUNICATERを通してＺ80からコールできます．

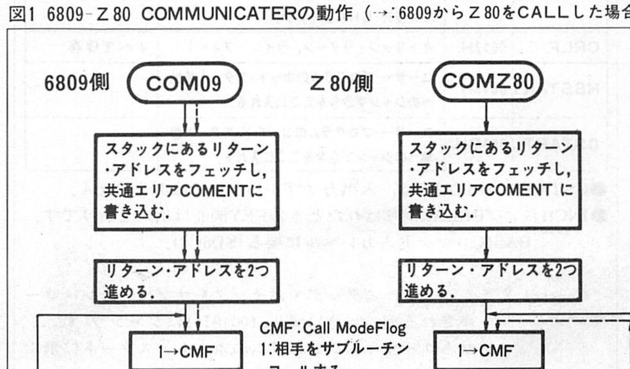
注1）1，0以外は書き込まないこと！特に80Hを書き込むと RESET をかけない限り動かなくなります．

注2）ただし，このプログラムを使わないでＺ80-6809の切り換えを行なうプログラムを実行した後ではできません．FM-8のＺ80用プログラムにはすべてこのCOMMUNICATERを使います．

COMMUNICATERの構成

COMMUNICATERは，6809からＺ80をコールするCOM09と，Ｚ80から6809をコールするCOMＺ80の2つのプログラムと共通のデータ・エリアからなっています．その動作を図1に示します．

図1 6809-Ｚ80 COMMUNICATERの動作（→:6809からＺ80をCALLした場合）



ここで6809からＺ80のサブルーチンをコールする場合の動きを点線で示します．このときCOM09によってコールされたＺ80のサブルーチンの中でCOMＺ80をコールし，6809のサブルーチンをコールすることは可能です．

したがって，6809がＺ80のサブルーチンをコールし，コールされたＺ80のサブルーチンが6809のサブルーチンをコールし，またその6809のサブルーチンがＺ80のサブルーチ

表1 入出力サブルーチン

名 称	ENTRY ADDR.	機 能	レジスタ
OUT CH	0000H	AccAをスクリーンまたはプリンタに出力.	すべて保存
IN CH	0003H	KEY入力データをAccAに入れる.	AccA以外保存
OUTLIN	0006H	Xレジスタで示されるアドレスからターミネータ(0DH)までを，スクリーンまたはプリンタに出力.	すべて保存
IN LIN	0009H	CRを押すまでKEY入力データを取り込み，Xレジスタの示すマドレスから順に並べる．最後は(0DH)最大入力文字数80字.	すべて保存
BREAK	000CH	KEY入力をチェックする．通常0→AccA	AccA以外保存
START	000FH	COMMUNICATERの初期化．画面の初期設定．に相当．その後001BHにジャンプ.	保存されない
CRLF	0012H	キャリッジ・リターン，ライン・フィード	すべて保存
HSSTART	0018H	ユーザー・プログラムのホット・スタート番地へのジャンプ命令をここに入れる.	
CSTART	001BH	ユーザー・プログラムのコールド・スタート番地へのジャンプ命令をここに入れる.	

●CSTART，HSTARTは，入出力サブルーチンではありません．
●INCHおよびBREAKが呼ばれたときのKEY機能は次のとおりです．
：BASICコマンド入力レベルに戻る($D8C2)．

Control X ：レジスタ，スタック・ポインタをサブルーチン・コールされる前の状態に戻し，(0018H)にジャンプ（ここからユーザー・プログラムのホット・スタートに飛ぶとよい）．
Control R ：プリンタON
Control Q ：プリンタOFF
ESCキーまたはSPACE KEY
：F-BACICでのESCKEYと同じ(BREAKルーチンのみ)．
●以上のプログラムはすべてBIOSを通じている（$FBFA)．

ンをコールするというようなことを繰り返すことも，それぞれのスタックが許す限り何重にもネスティングできます．

使用例

表2にこのCOMMUNICATERの使用例を示します．

COM09，COMＺ80の2つはどちらも相手側プロセッサのサブルーチン・エントリ・アドレスをCOM09，COMＺ80をコールした命令の次の２バイトから取ることになっているので，**JSR$6E45またはCALL6E00H，REST38H**をオペレートとし，相手側プロセッサのサブルーチン・エントリ・アドレス２バイトをオペランドとする１つのサブルーチン・コール命令と見ることができるようになっています．そしてサブルーチン実行後は，オペランドにあたる２バイトのデータの次の命令のアドレスに戻ってくるようになっています．

データの受け渡し

プロセッサ間のデータの受け渡しは，AccA（6809）－AccA（Ｚ80)，X（6809）－ＤＥ（Ｚ80）の各レジスタを通して行なわれます．その使用例を表３に示します．

表2 COM09およびCOMZ80の使用例

（1）6809からＺ80をコールする．

```
BD6E45  JSR  $6E45  COM09のエントリ・アドレス
5A 80   FDB  $5A80  Z80のサブルーチンのエントリ・アドレス
```

（2）Ｚ80から6809をコールする(RST38Hを使用したとき)．

```
FF      RST   38H   COMZ80へ
09 68   DEFW  6809H 6809サブルーチンのエントリ・アドレス
```

（3）Ｚ80から6809をコールする(CALL命令を使用したとき)．

```
CD006E  CALL  6E00H COMZ80のエントリ・アドレス
09 68   DEFW  6809H 6809サブルーチンのエントリ・アドレス
```

表3 Ｚ80から入出力サブルーチンをコールする例

（1）AccAを使って6809にデータを送る(１文字出力)．

```
LD    A,41H  出力データ"A"
RST   38H
DEFW  OUTCH  1文字出力ルーチン・エントリ・アドレス
RET
```

（2）DEレジスタ・ペアを用いる(BIOSをコール)．

```
LD    DE,RCB FM-8 Reguest Control Bsueth
RST   38H
DEFW  BIOS       BIOS EQU $F2D8(ROM BASIC時)
RET
```

表4 Ｚ80ディスアセンブラの変更点

アドレス	変 更 前	変 更 後
506B	6E6A	6E72
5C18	86C39700	31006E21
5C1C	9738308C	030018AB
5C20	CA1F101E	86C39700
5C24	89DD0130	9738CC6A
5C28	88C31F10	6EDD01CC
5C3C	1E89	006E

以上のことから，このCOMMUNICATERを使うと，ただ単にサブルーチンをコールするだけのようにして，6809からＺ80を，Ｚ80から6809をコールすることができ，プログラムの作成が簡単に行なえます．

ユーザー・プログラムとのリンク方法

このCOMMUNICATERと入出力ルーチンをユーザー・プログラムとリンクして使う場合について説明します．

ソース・リストに示したプログラムは，COMMUNICATERと入出力ルーチンをまとめてロードするようになっています．それは入出力ルーチンの部分がリロケータブルであるためで，あいている所にロードできます．

したがって，表１のエントリ・アドレスは，ロードしたアドレスの先頭からのオフセットになります．

リロケータブルでないCOMMUNICATERの部分は，STARTルーチンを起動することによって所定の位置（6E00H～6E70H）に移されます．

また，STARTルーチンはその後，画面の初期化を行ない，入出力ルーチンの先頭から001BHオフセットされたアドレスにジャンプするので，そこにユーザー・プログラ

ムのコールド・スタート・アドレスへのジャンプ命令を書き込んでおけばよいことになります．

また0018Hにユーザー・プログラムのホット・スタート・アドレスへのジャンプ命令を書き込んでおけば（INCH，BREAKルーチンのKEY入力機能を利用することにより），ホット・スタートができます．

ユーザー・プログラムがＺ80のプログラムの場合は，COM09を通して，Ｚ80プログラムのホット・スタート，コールド・スタートをコールするようなプログラムを作成，そこへジャンプするようにすればよいでしょう．

またCOM09，COMＺ80をそれだけユーザー・プログラムとリンクして使う場合，リセット時とリスタート時の（Ｚ80の）ジャンプ命令を書き込んでおくイニシャライズ・ルーチンが必要になります．その設定値は，

0000　C36A6E	JP　6E6AH	RESET時
0038　C3006E	JP　6E00H	RESTART 38H時

この２つのジャンプ命令です．したがってこれを書き込んでおいてください．

おわりに（注意点）

前回発表したＺ80セルフ・ディスアセンブラにも，このCOMMUNICATERと同様のものを使っていますが，ディス・アセンブラのＩ/Ｏプログラム（転送されたもの）と今回のCOMMUNICATERでアドレスが重なっている部分があるため，このコミュニケータを使ったプログラムとディスアセンブラを同時には使えません．同時に使うためにはディスアセンブラの変更を表４のように行なってください．

6809コミュニケーション プログラム・リスト

```
                             NAM    COM09
6E45                         ORG    $6E45
                        *
                        * Communication program ( M6809 )
                        *
6E45 34   6D            COM09   PSHS   U,Y,DP,B,CC
6E47 EE   67                    LDU    7,S       fetch return address
6E49 37   20                    PULU   Y         fetch entry addr. ( Z-80 )
6E4B EF   67                    STU    7,S       restore return addr.
6E4D 109F 07                    STY    <COMENT   set entry ( to Z-80 )
6E50 C6   01                    LDB    #1        set CMF  ( 1:call mode )
6E52 CE   0007          SETZ80  LDU    #COMENT   point common area
6E55 36   16                    PSHU   X,B,A     exchange registers and CMF
                        *                            ( X-DE,A-A,B=CMF )
6E57 C6   01                    LDB    #1        to Z80 ( CNTZ80=1 )
6E59 F7   FD05                  STB    CNTZ80
                        *
                        * to Z80
                        *------------------------------------------
                        * from Z80
                        *
6E5C 12                         NOP
6E5D 37   16                    PULU   X,B,A     exchange registers and CMF
6E5F 5D                         TSTB             test CMF
6E60 27   05   6E67             BEQ    RETZ80
                        *
                        * If called by Z80
                        *
6E62 AD   D4                    JSR    [,U]      jump to subroutine ( M6809 )
6E64 5F                         CLRB             clear CMF  ( 0:return )
6E65 20   EB   6E52             BRA    SETZ80    return to Z80
                        *
                        * Return from subroutine ( Z80 )
                        *
6E67 35   ED            RETZ80  PULS   PC,U,Y,DP,B,CC
                        *
                        * Common area  ( M6809-Z80 )
                        *
          0003          COMA    EQU    3         AccA
          0004          CMF     EQU    4         call mode flag
          0005          COMIX   EQU    5         X-register
          0007          COMENT  EQU    7         entry of subroutine
          FD05          CNTZ80  EQU    $FD05     control  Z80-M6809  ( bit0 )
                                END
TOTAL ERRORS 00000--00000
TOTAL WARNINGS 00000--00000
```

Z80コミュニケーション プログラム・リスト

```
                        ;Z80 to M6809 communicater (Z80)
                        ;
                                ORG     6E00H
6E00 E3         COMZ80: LD      (SP),HL         ;fetch return addr.
6E01 C5                 PUSH    BC
6E02 46                 LD      B,(HL)          ;fetch entry addr. ( M6809 )
6E03 23                 INC     HL
6E04 4E                 LD      C,(HL)
6E05 23                 INC     HL
6E06 ED430700           LD      (COMENT),BC     ;set entry addr. ( M6809 )
6E0A C1                 POP     BC              ;restore return address
6E0B E3                 LD      (SP),HL
6E0C E5                 PUSH    HL
6E0D 210600             LD      HL,COMIX+1      ;point common area
6E10 73                 LD      (HL),E          ;exchange E ( -XL )
6E11 5D                 LD      E,L             ;set CMF ( =6 not 0 :call mode )
6E12 2D         SET09:  DEC     L
6E13 72                 LD      (HL),D          ;exchange D ( -XH )
6E14 2D                 DEC     L
6E15 73                 LD      (HL),E          ;CMF
6E16 2D                 DEC     L
6E17 77                 LD      (HL),A          ;exchange A ( -A )
6E18 AF                 XOR     A               ;to M6809 ( CNTZ80=0 )
6E19 3205FD             LD      (CNTZ80),A
                        ;
                        ; to M6809
                        ;-----------------------------------
                        ; from M6809
                        ;
6E1C 00                 NOP
6E1D 7E         ONREST: LD      A,(HL)          ;exchange A ( -A )
6E1E 2C                 INC     L
6E1F 5E                 LD      E,(HL)          ;CMF
6E20 2C                 INC     L
6E21 56                 LD      D,(HL)          ;exchange D ( -XH )
6E22 2C                 INC     L
6E23 1D                 DEC     E               ;test CMF ( 1 or 0 )
6E24 5E                 LD      E,(HL)          ;exchange E ( -XL )
6E25 C2426E             JP      NZ,RET09
                        ;
                        ; If called by M6809
                        ;
6E28 C5                 PUSH    BC
6E29 DDE5               PUSH    IX
6E2B FDE5               PUSH    IY
6E2D 2C                 INC     L
6E2E 46                 LD      B,(HL)          ;fetch entry address ( Z80 )
6E2F 2C                 INC     L
6E30 6E                 LD      L,(HL)
6E31 60                 LD      H,B
6E32 CD446E             CALL    JPHL            ;( jump to subroutine )
                        ;
6E35 FDE1               POP     IY
6E37 DDE1               POP     IX
6E39 C1                 POP     BC
6E3A 210600             LD      HL,COMIX+1      ;point common area
6E3D 73                 LD      (HL),E          ;exchange E ( -XL )
6E3E 5C                 LD      E,H             ;clear CMF ( 0:return mode )
6E3F C3126E             JP      SET09           ;return to M6809
                        ;
                        ; Return from subroutine ( M6809 )
                        ;
6E42 E1         RET09:  POP     HL
6E43 C9                 RET
                        ;
                        ;
6E44 E9         JPHL:   JP      (HL)            ;jump to subroutine ( M6809 )
                        ;
                        ; Common area ( Z80-M6809 )
                        ;
                COMA:   EQU     3               ;AccA
                CMF:    EQU     4               ;call mode flag
```

Z80コミュニケーション プログラム・リスト

```
                    COMIX:   EQU    5           ;DE - X reg.
                    COMENT:  EQU    7           ;entry of subroutine
                    CNTZ80:  EQU    FD05H       ;control  Z80-M6809


                    ;
                    ;
                    ; Initialize routine
                    ;
                             ORG    6E6AH
                    ;
6E6A 31006E         INITZ:   LD     SP,6E00H    ;set stack pointer
6E6D 210300                  LD     HL,COMA     ;point common area
6E70 18AB                    JR     ONREST
                    ;
                             END
```

FM-8入出力 プログラム＋6809-Z80コミュニケータ プログラム・リスト

```
                                NAM    IOZ80
                          *
                          * Input-Output Program for FM-8
                          *     with M6809-Z80 communicater
                          *
                          *
 0000                           ORG    $0000    position independent
                          *
 0000 16   0060 0063 OUTCH  LBRA   OUTCHR   output character ( AccA )
 0003 16   0094 009A INCH   LBRA   INCHR    input character ( AccA )
 0006 16   0065 006E OUTLIN LBRA   OUTLIR   output line
 0009 16   0012 001E INLIN  LBRA   INLINR   input line
 000C 16   0091 00A0 BREAK  LBRA   BRAZBR   break check
 000F 16   0101 0113 START  LBRA   INIT     initialize I-O program
 0012 16   0042 0057 CRLF   LBRA   CRLFR    CR LF
 0015 16   00FB 0113 DUMMY  LBRA   INIT
 0018 16   01F2 020D HSTART LBRA   HSTUSR   to hot start ( usre program )
 001B 16   01E9 0207 CSTART LBRA   CSTUSR   to cold start ( usre program )
                          *
                          * Input one Line
                          *     data pointer = X reg.
                          *     delimiter = $0D
                          *
 001E 34   57        INLINR PSHS   U,X,B,A,CC
 0020 17   0125 0148        LBSR   COLOR    set color and entry position
 0023 86   17               LDA    #$17     cursor on
 0025 17   0134 015C        LBSR   CURSOR   :
 0028 33   0A               LEAU   10,X     buffer
 002A CC   0054             LDD    #84      maximun input character
 002D ED   06               STD    6,X
 002F 34   40               PSHS   U
 0031 CE   0000             LDU    #0       no mesage
 0034 86   12               LDA    #18      LINE INPUT request number
 0036 17   0119 0152        LBSR   PRINT    call BIOS ( LINE INPUT )
 0039 32   62               LEAS   2,S
 003B E6   05               LDB    5,X      number of character
 003D 33   44               LEAU   4,U      delete location information
 003F C0   03               SUBB   #3
 0041 AE   63               LDX    3,S
 0043 5A             INLL   DECB            transfer to usre buffer ( X )
 0044 2F   06   004C        BLE    LLEN     :
 0046 A6   C0               LDA    ,U+      :
 0048 A7   80               STA    ,X+      :
 004A 20   F7   0043        BRA    INLL     :
 004C 86   0D        LLEN   LDA    #$D      delimiter
 004E A7   84               STA    ,X
 0050 86   16               LDA    #$16     cursor off
 0052 17   0107 015C        LBSR   CURSOR   :
 0055 35   D7               PULS   PC,U,X,B,A,CC
                          *
                          * CR and LF
                          *
```

FM-8入出力 プログラム+6809↔Z80コミュニケータ プログラム・リスト

```
0057 34    03            CRLFR   PSHS  A,CC
0059 86    0A                    LDA   #$A    LF code
005B 8D    06   0063             BSR   OUTCHR
005D 86    0D                    LDA   #$D     CR code
005F 8D    02   0063             BSR   OUTCHR
0061 35    83                    PULS  PC,A,CC
                         *
                         * Output one Character ( AccA )
                         *
0063 34    57            OUTCHR  PSHS  U,X,B,A,CC
0065 30    61                    LEAX  1,S     point data
0067 CE    0001                  LDU   #1      number of data
006A 34    10                    PSHS  X
006C 20    11   007F             BRA   ENCNT
                         *
                         * Output one Line
                         *     data pointer = X reg.
                         *     delimiter = $0D
                         *
006E 34    57            OUTLIR  PSHS  U,X,B,A,CC
0070 C6    0D                    LDB   #$D     delimiter
0072 34    10            OUTL2   PSHS  X
0074 CE    0000                  LDU   #0
0077 E1    80            DATCNT  CMPB  ,X+     count data
0079 27    04   007F             BEQ   ENCNT   :
007B 33    41                    LEAU  $01,U   :
007D 20    FB   0077             BRA   DATCNT  :
007F 34    40            ENCNT   PSHS  U
0081 17    00C4 0148             LBSR  COLOR   set color
0084 EE    E1                    LDU   ,S++
0086 27    0E   0096             BEQ   NOUT    no character
0088 86    14                    LDA   #20     OUT-SCREEN request number
008A 17    00C5 0152             LBSR  PRINT   print ( screen )
008D 6D    08                    TST   8,X     printer flag on ?
008F 27    05   0096             BEQ   NOUT
0091 86    0E                    LDA   #14     OUT-PRINTER request number
0093 17    00D7 016D             LBSR  MEND2   print ( printer )
0096 32    62            NOUT    LEAS  2,S
0098 35    D7                    PULS  PC,U,X,B,A,CC
                         *
                         * Input one Character ( AccA )
                         *
009A 34    55            INCHR   PSHS  U,X,B,CC
009C C6    01                    LDB   #1      AccB > 0 : wait keyin
                                                 and put keydata in AccA
009E 20    03   00A3             BRA   INKEY
                         *
                         * Break Check
                         *
00A0 34    55            BRAZBR  PSHS  U,X,B,CC
00A2 5F                          CLRB          AccB = 0 : key scan
                         *
                         * Inkey
                         *
00A3 86    16            INKEY   LDA   #$16    $16 : cursor off
00A5 5D                          TSTB
00A6 2F    01   00A9             BLE   INK3    no cursor ( on Break )
00A8 4C                          INCA          $17 : cursor on
00A9 17    00B0 015C     INK3    LBSR  CURSOR  cursor control
00AC 6D    09            INKEY2  TST   9,X     test escape flag
00AE 27    04   00B4             BEQ   INK1
00B0 5D                          TSTB
00B1 2E    01   00B4             BGT   INK1
00B3 53                          COMB          AccB < 0 : wait keyin
00B4 33    E3            INK1    LEAU  ,--S    buffer
00B6 4C                  INK11   INCA
00B7 26    FD   00B6             BNE   INK11
00B9 86    15                    LDA   #21     INKEY request number
00BB 17    00AD 016B             LBSR  MEND    call BIOS ( INKEY )
00BE 7D    0313                  TST   TSTOP   stop key check
00C1 27    06   00C9             BEQ   NOABRT
00C3 4F                  ABEN    CLRA          clear DP register
00C4 1F    8B                    TFR   A,DP    :
```

```
00C6 7E    D8C2              JMP   ABORT    to stop routine on FM-8 ( ROM
00C9 5D            NOABRT TSTB
00CA 27    04   00D0         BEQ   KEYEXT
00CC 6D    41                TST   1,U      push key ?
00CE 27    E6   00B6         BEQ   INK11    no : wait keyin
00D0 A6    E1      KEYEXT LDA   ,S++     get input data
00D2 81    03                CMPA  #3       control C ?
00D4 26    05   00DB         BNE   NOCTRC
00D6 7C    0313              INC   TSTOP
00D9 20    E8   00C3         BRA   ABEN     to stop routine
00DB 81    11      NOCTRC CMPA  #$11     control Q ?
00DD 26    02   00E1         BNE   NOCTRR
00DF 6F    08                CLR   8,X      clear printer flag
00E1 81    12      NOCTRR CMPA  #$12     control R ?
00E3 26    02   00E7         BNE   NOCTRQ
00E5 A7    08                STA   8,X      set printer flag
00E7 81    18      NOCTRQ CMPA  #$18     control X ?
00E9 26    05   00F0         BNE   NOCTRX
00EB 35    55                PULS  U,X,B,CC
00ED 16    FF28 0018         LBRA  HSTART   to hot start of usre program
00F0 5D            NOCTRX TSTB
00F1 2F    0A   00FD         BLE   NOINRQ   no input request
00F3 34    02                PSHS  A
00F5 86    16                LDA   #$16     cursor off
00F7 8D    63   015C         BSR   CURSOR   :
00F9 35    02                PULS  A
00FB 20    12   010F         BRA   INRQ
00FD 81    20      NOINRQ CMPA  #$20     space key ?
00FF 27    04   0105         BEQ   SPC
0101 81    1B                CMPA  #27      escape key ?
0103 26    09   010E         BNE   NOESC
0105 4F            SPC       CLRA
0106 6D    09                TST   9,X      test escape flag
0108 26    07   0111         BNE   KEYEND
010A 6C    09                INC   9,X      set escape flag
010C 20    9E   00AC         BRA   INKEY2   wait key input
010E 4F            NOESC     CLRA
010F 6F    09      INRQ      CLR   9,X      clear escape flag
0111 35    D5      KEYEND    PULS  PC,U,X,B,CC
                   *
                   * Initialize I-O program
                   *
0113 33    8C 5F   INIT      LEAU  <WBUF,PCR
0116 8D    49   0161         BSR   CALSUB   initialize console
0118 33    4B                LEAU  11,U
011A 8D    45   0161         BSR   CALSUB   define KEY(5)
011C 6F    08                CLR   8,X      clear printer flag
011E 6F    09                CLR   9,X      clear escape flag
0120 86    07                LDA   #7       color code ( white )
0122 B7    01E5              STA   COLRCD   set color
                   *
                   * Transfer communicate program to 6E00H-6E71H
                   *
0125 30    8C 6D             LEAX  <COMZ80,PCR source
0128 CE    6E00              LDU   #COMTFR destination
012B A6    80      MOVLP     LDA   ,X+
012D A7    C0                STA   ,U+
012F 1183  6E72              CMPU  #COMEND
0133 26    F6   012B         BNE   MOVLP
                   *
0135 86    C3                LDA   #$C3     JP code for Z80
0137 97    00                STA   <0       reset
0139 97    38                STA   <38H     restart 38H
013B CC    006E              LDD   #$006E   entry of COMZ80 ( 6E00H )
013E DD    39                STD   <39H
0140 CC    6A6E              LDD   #$6A6E   entry of init. for Z80
0143 DD    01                STD   <1
0145 16    FED3 001B         LBRA  CSTART
                   *
                   * Set color
                   *
0148 33    8C 3F   COLOR     LEAU  <COLBUF,PCR
014B B6    01E5              LDA   COLRCD   color code
```

FM-8入出力 プログラム+6809↔Z80コミュニケータ プログラム・リスト

```
014E A7   45             STA    5,U
0150 20   0F   0161      BRA    CALSUB
                  *
                  * Print or line input
                  *
0152 30   8D 00BA   PRINT  LEAX   RCB,PCR
0156 EF   04             STU    4,X        set number of data
0158 EE   62             LDU    2,S
015A 20   0F   016B      BRA    MEND
                  *
                  * Cursor on / off
                  *
015C 33   8C 31     CURSOR LEAU   <BUFCSR,PCR
015F A7   43             STA    3,U
                  *
                  * Call sub CPU
                  *
0161 30   8D 00AB   CALSUB LEAX   RCB,PCR    request control block
0165 86   10             LDA    #16        sub-out request number
0167 6F   04             CLR    4,X
0169 A7   05             STA    5,X        set number of data
016B EF   02        MEND   STU    2,X        set buffer address
016D A7   84        MEND2  STA    ,X         set request number
016F 6F   01             CLR    1,X        clear status
0171 6E   9F FBFA        JMP    [BIOS]     jump to BIOS
                  *
                  * WIDTH 40,25 : CONSOLE 0,25,0,0
                  *
0175      0A        WBUF   FCB    $A
0176      00             FCB    0
0177      01             FCB    1          initialize console code
0178      00             FCB    0
0179      28             FCB    40         40 character / line
017A      19             FCB    25         25 line / page
017B      00             FCB    0          scrolle top
017C      19             FCB    25         scrolle bottom
017D      00             FCB    0          display PFKEY switch
                                             ( 0:off  2:on )
017E      01             FCB    1
017F      00             FCB    0          console color switch
                                             ( 0:normal  $FF:green only
                  *
                  *
                  * KEY5,"EXEC"+CHR$(13)
                  *
0180      0A        KYBUF  FCB    $A
0181      00             FCB    0
0182      2A             FCB    $2A        define key code
0183      05             FCB    5          key number
0184      05             FCB    5          number of character
0185      45             FCC    'EXEC'
0186      58
0187      45
0188      43
0189      0D             FCB    $D         CR code
                  *
                  * Put strings
                  *
018A      0A        COLBUF FCB    $A
018B      00             FCB    0
018C      03             FCB    3          put strings code
018D      02             FCB    2          number of character
018E      11             FCB    $11        DC1 : set color
018F      07             FCB    7          color code ( white )
                  *
0190      0A        BUFCSR FCB    $A,0
0191      00
0192      0C             FCB    $C         control console code
0193      16             FCB    $16        cursor switch control
0194      02             FCB    2
                  *
                  *
                  * COMZ80 Object program ( Z80 )
```

```
                                     *
0195   E3           COMZ80 FCB       $E3        LD       (SP),HL
0196   C5                  FCB       $C5        PUSH     BC
0197   46                  FCB       $46        LD       B,(HL)
0198   23                  FCB       $23        INC      HL
0199   4E                  FCB       $4E        LD       C,(HL)
019A   23                  FCB       $23        INC      HL
019B   ED43                FDB       $ED43      LD       (COMENT),BC
019D   0700                FDB       $0700
019F   C1                  FCB       $C1        POP      BC
01A0   E3                  FCB       $E3        LD       (SP),HL
01A1   E5                  FCB       $E5        PUSH     HL
01A2   21                  FCB       $21        LD       HL,COMIX+1
01A3   0600                FDB       $0600
01A5   73                  FCB       $73        LD       (HL),E
01A6   5D                  FCB       $5D        LD       E,L
01A7   2D           SET09  FCB       $2D        DEC      L
01A8   72                  FCB       $72        LD       (HL),D
01A9   2D                  FCB       $2D        DEC      L
01AA   73                  FCB       $73        LD       (HL),E
01AB   2D                  FCB       $2D        DEC      L
01AC   77                  FCB       $77        LD       (HL),A
01AD   AF                  FCB       $AF        XOR      A
01AE   32                  FCB       $32        LD       (CNTZ80),A
01AF   05FD                FDB       $05FD
01B1   00                  FCB       $00        NOP
01B2   7E           ONREST FCB       $7E        LD       A,(HL)
01B3   2C                  FCB       $2C        INC      L
01B4   5E                  FCB       $5E        LD       E,(HL)
01B5   2C                  FCB       $2C        INC      L
01B6   56                  FCB       $56        LD       D,(HL)
01B7   2C                  FCB       $2C        INC      L
01B8   1D                  FCB       $1D        DEC      E
01B9   5E                  FCB       $5E        LD       E,(HL)
01BA   C2                  FCB       $C2        JP       NZ,RET09
01BB   426E                FDB       $426E
01BD   C5                  FCB       $C5        PUSH     BC
01BE   DDE5                FDB       $DDE5      PUSH     IX
01C0   FDE5                FDB       $FDE5      PUSH     IY
01C2   2C                  FCB       $2C        INC      L
01C3   46                  FCB       $46        LD       B,(HL)
01C4   2C                  FCB       $2C        INC      L
01C5   6E                  FCB       $6E        LD       L,(HL)
01C6   60                  FCB       $60        LD       H,B
01C7   CD                  FCB       $CD        CALL     JPHL
01C8   446E                FDB       $446E
01CA   FDE1                FDB       $FDE1      POP      IY
01CC   DDE1                FDB       $DDE1      POP      IX
01CE   C1                  FCB       $C1        POP      BC
01CF   21                  FCB       $21        LD       HL,COMIX+1
01D0   0600                FDB       $0600
01D2   73                  FCB       $73        LD       (HL),E
01D3   5C                  FCB       $5C        LD       E,H
01D4   C3                  FCB       $C3        JP       SET09
01D5   126E                FDB       $126E
01D7   E1           RET09  FCB       $E1        POP      HL
01D8   C9                  FCB       $C9        RET
01D9   E9           JPHL   FCB       $E9
                    *
                    * Communication program ( M6809 )
                    *
01DA 34    6D       COM09  PSHS      U,Y,DP,B,CC
01DC EE    67              LDU       7,S        fetch return address
01DE 37    20              PULU      Y          fetch entry addr. ( Z-80 )
01E0 EF    67              STU       7,S        restore return addr.
01E2 109F  07              STY       <COMENT    set entry ( to Z-80 )
01E5 C6    01              LDB       #1         set CMF  ( 1:call mode )
01E7 CE    0007     SETZ80 LDU       #COMENT    point common area
01EA 36    16              PSHU      X,B,A      exchange registers and CMF
                    *                                   ( X-DE,A-A,B=CMF )
01EC C6    01              LDB       #1         to Z80 ( CNTZ80=1 )
01EE F7    FD05            STB       CNTZ80
```

FM-8入出力 プログラム+6809↔Z80コミュニケータ プログラム・リスト

```
                                 *
                                 * to Z80
                                 *-----------------------------------------
                                 * from Z80
                                 *
          01F1 12                     NOP
          01F2 37     16              PULU    X,B,A     exchange registers and CMF
          01F4 5D                     TSTB              test CMF
          01F5 27     05    01FC      BEQ     RETZ80
                                 *
                                 * If called by Z80
                                 *
          01F7 AD     D4              JSR     [,U]      jump to subroutine ( M6809 )
          01F9 5F                     CLRB              clear CMF  ( 0:return )
          01FA 20     EB    01E7      BRA     SETZ80    return to Z80
                                 *
                                 * Return from subroutine ( Z80 )
                                 *
          01FC 35     ED         RETZ80 PULS  PC,U,Y,DP,B,CC
                                 *
                                 *
                                 * Initialize Z80  ( object program )
                                 *
          01FE 12                     NOP
          01FF        31         INITZ  FCB     $31       LD      SP,6E00H
          0200        006E              FDB     $006E
          0202        21                FCB     $21       LD      HL,CMF
          0203        0300              FDB     $0300
          0205        18AB              FDB     $18AB     JR     ONREST
                                 *
                                 * Common area  ( M6809-Z80 )
                                 *
                      0003       COMA   EQU     3         AccA
                      0004       CMF    EQU     4         call mode flag
                      0005       COMIX  EQU     5         X-register
                      0007       COMENT EQU     7         entry of subroutine
                      FD05       CNTZ80 EQU     $FD05     control  Z80-M6809  ( bit0 )
                                 *
                                 *
                                 * Dummy program for cold start
                                 *    5100H ニ アル Z80 ノ プログラム ニ ジャンプ
                                 *
          0207 BD     6E45       CSTUSR JSR     $6E45
          020A        5100              FDB     $5100
          020C 39                       RTS                 ( イラナイ )
                                 *
                                 * Dummy program for hot start
                                 *    control X ヲ オスト AccA=1  : ソノホカ ノ トキ AccA=0
                                 *
          020D 86     01         HSTUSR LDA     #1
          020F 39                       RTS
                                 *
                                 *
                                 *
          0210        0008       RCB    RMB     8         request control brock
          0218        0001       PRTF   RMB     1         printer flag
          0219        0001       ESCFLG RMB     1         escape flag
          021A        0054       BUFLIN RMB     84        buffer
                      01E5       COLRCD EQU     $1E5      color code
PAGE  008 (           ,002106)  IDZ80

                      0313       TSTOP  EQU     $313      break status
                      D8C2       ABORT  EQU     $D8C2     stop routine for FM-8  ( ROM )
                      FBFA       BIOS   EQU     $FBFA
                      6E00       COMTFR EQU     $6E00
                      6E72       COMEND EQU     $6E72
                                        END
TOTAL ERRORS 00001--00130
TOTAL WARNINGS 00004--00190
```

6809でZ80ソフトを！

Z80 System Monitor

FM-8 ■ETF Laboratory 笠作貴弥/天内 潤

私は，長い間自作のZ80システムを使ってきましたが，最近のソフト・ハードの進歩には太刀打ちできず，メインCPUを6809からZ80へ切り換えることのできるFM-8を購入しました．

しかし，現時点では本体とZ80ソフトカードのみでは，モニタ・プログラムがないため，Z80のソフトを走らすことができません．

そこで開発したのがこのプログラムです．普通のモニタでは，つまらないので，MZ-80のSP-1002に仕様を似せました．これにより，システムにあまり依存しないプログラム（アセンブラなど）の移植が容易にできます!!

Z80ソフトカード

Z80ソフトカードの使用方法や切り換えの方法は，マニュアルやI/O'82年2月号に詳しく説明されているので，省略しますが，ソフトカードのクロックについて調べてみました．

初めZ80のAバージョンを使用しているので，4MHzで動作すると思ったのですが，簡単なプログラムを走らせて調べてみると，2.5MHz前後であることが解りました．6809のメイン・クロックの原発が約5MHzなので，その1/2であると思います．

最高速で動作しないのは残念ですが，それ以上にZ80↔6809の切り換えや画面への表示にかなり時間がかかるようなので，たとえ4MHzで動作してもあまり変わらないのではないかと思います．

プログラムの入力

MON RET でモニタを呼び出して，入力してください．＄900～＄FFFまでの1,792バイトです．入力し終ったらすぐセーブしましょう．SAVEM"Z-80 MON", &H900,&HFFF, &H900.

プログラム

プログラムは大きく分けて，次の3つの部分から成り立っています（メモリ・マップ参照）．

❶Z80ドライバー

6809のマシン語によって書かれているこの部分は，CPU切り換えルーチン，BIOSコール・ルーチンおよびCMT関係の処理ルーチンから構成されています．

この部分は文書を読むより，**フローチャート1**およびアセンブル・リストを見て頂いた方が解り易いと思いますが，少し説明しておくと，まずF-BASICより起動(EXEC&H900)されるとZ80のスタート・ベクトルを設定し，CPUをZ80へ切り換えます．Z80側の処理が終って，再び6809に変わると，ワーク・エリア（RETFLG ＄8FD）の値を調べて，各処理ルーチンへジャンプします．その後は再びZ80へ切り換え，RETFLGの値が1になるまで繰り返します．

❷IOCS&サブルーチン・パッケージ

Z80のマシン語によって書かれているこの部分は，Z80のユーザープログラムから自由にコールすることができます．CPUの切り換え，サブシステムのコントロールなどをまったく意識する必要はありません．

各サブルーチンの仕様は，SP-1002に似せてありますが，ハードウェアが異なるので，完全コンパチブルすることはできないし，すべての代替サブルーチンを内臓させることができませんでした．

基本的なものは，一応揃えてあるので，他のものはユーザー側で用意してください（**表1**）．

以下少し特別なサブルーチンについて述べます．

PRNT：1文字プリントのサブルーチンですが，MZとコンパチにするため，ODHが送られてくるとLETNLを実行します．その他の制御コードについては，FM-8と同じです．

?WR1：?WRDのみでセーブするようにしたのでダミーのサブルーチンです．

?RD1：ファイル名のみロードして，10F1Hから8バイトにストア．

GETL：スクリーン・エディタを含めた1行入力のサブルーチンですが，必ず変更されないとデータは転送されません．つまり，カーソルを他の行へ移動して RET してもその行の文字列を変更しない限り，DEレジスタで示されるバッファの先頭に ESC （1BH）が返えされるのみです．注意してください．

GETC：非常に面白いサブルーチンで，GETLで入力した行以外で変更されたエリア（エディットした行）があれば(DE)以降に転送されます．変更されたエリアがない場合は先頭に ESC が入ります．

このサブルーチンのおかげで，F-BASICや本モニタのメモリ・ダンプの同時複数行エディットが可能なわけです．具体的使用方法は，アセンブル・リストのEDITのところを見てください．

WIDTH：ワーク・エリア（表2）の値に従って，画面モードの設定を行ないます．ユーザープログラムからは，ワーク・エリアに適当な値をセットして，ここをコールすれば，画面がセットされます．

カラースイッチを白黒モードにセットするとスクロールの速さが速くなります．

❸簡易モニタ

IOCSとサブルーチン・パッケージだけでは仕方がないので，簡易モニタを内臓させました．サイズを小さくするために機能は最小限とし，エラーチェックなどは省いてあるので注意してください．

以下各コマンドの説明です．

Mコマンド　メモリ・チェンジ

書式　M␣アドレス RET

16進2桁を入力すると，アドレスがインクリメントします．RET で次のアドレスへ SP でバックします．

Dコマンド　メモリ・ダンプ

書式　D␣アドレス RET

指定したアドレスよりメモリ・ダンプします．ESC で一時停止，CTRL C でコマンド待ちになり，エディットが可能です．

エディットは，自由にカーソルを移動して，一度に複数行のエディットが可能です（1行ごと RET する必要がなく最後に1回だけでよい）．

Gコマンド

書式　G␣アドレス RET

指定されたアドレスにジャンプします．

Jコマンド

書式　J RET

1106Hに書かれたアドレスにジャンプします．テープのロード後このコマンドにて，ユーザープログラムへジャンプすることができます．

Wコマンド　セーブ

書式　W␣先頭アドレス，最終アドレス，実行開始アドレス，ファイル名 RET

F-BASICと同じフォーマットで記録します．

Lコマンド　ロード

書式　L RET

ファイル名は指定できませんが，ファイルを見つけるごとに一時停止して，ロードするか聞いてくるので，ロードするなら Y ，スキップするなら S を押してください．他のキーを入力するとコマンド待ちになります．

@コマンド　モーターON

書式　@ RET

F-BASICのMOTOR　ONと同じです．

!コマンド

書式　! RET

F-BASICのホット・スタートへジャンプします．

＊注意：MDGWコマンドに共通のことですが，アドレスは必ず16進4桁で入力してください．

ユーティリティ

以上のプログラムを有効に使ってもらうために次の2つのプログラムを作りました．

❶MZテープ・ローダ

このプログラムは，MZ-80K/Cでセーブされたテープを FM-8上にロードするためのプログラムです．

プログラムの内容は，**フローチャート2**，**表3**を見てもらえば解ると思うので，使用方法を説明します．

プログラムをスタートさせると（EXEC&H7000）“Searching MZ Tepe”と表示するので，MZのテープをレコーダにセットして，PLAYボタンを押してください．しばらくすると，ファイル名などのファイル情報を表示し，データ・ブロックのロードに入ります．

何もなければ，“No Errors”と表示しプログラムを終了します．途中，エラーがあればエラー表示か無限ループに入り込むので，STOP RESET で最初からやりなおしてください．何度やってもダメな場合は，$70F1の値を調整して見てください．

私の場合は，$48～$50の間で，うまくロードできました．

レコーダ側の調整やテープの種類の選択もお忘れなく．

このプログラムはMZ側でセーブされたのと同じアドレスにロードするようになっていますが，**表4**のアドレスの内容を変更することで，任意のアドレスからロードすることができます．

以上ですが，MZのプログラムを移植するときや，データなどを転送する場合に便利なプログラムだと思います．

❷サブルーチン・コンバータ

MZのプログラムのモニタをコールしている部分を本プログラムのエントリ・アドレスに自動変換するプログラムです．

プログラムは，リストを見てもらえば，解ると思いますが，CDHまたは，C3Hに続く2バイトが，MZのエントリ・アドレスとして登録されている場合，それに代わる本プログラムのサブルーチンのアドレスに無条件で書き換えています．

以上の条件を満たせばデータであろうが，プログラムであろうが，勝手に書き換えてしまうので，注意が必要です．

MZのプログラムの移植テクニック

私は，シャープのエディタ・アセンブラやBASEなどを移植して使っています．

一般に他の機種のプログラムを移植する場合，そのプログラムの動作を確実に理解することが必要です．それには，前にあげた，サブルーチン・コンバータやディス・アセンブラを利用します．また，両機種の特性やハードウェア，ソフトウェアの違いを充分に理解することも必要です．

本プログラムでは，IOCSなどの仕様をMZに似せることであまりそれらを意識せずにできるようになっていますが，ハードウェアを直接アクセスするようなプログラム（ゲームなど）は，ほとんど無理なのであきらめるか，そのプログラムのアルゴリズムのみ利用して最初から作りなおした方がいいと思います．

ここで具体的な例として，BASEの移植について述べると，まずI/O 8月号のダンプ・リストを入力するなり，テープを購入するなりして，FM-8上にロードします．

次にサブルーチン・コンバータによって，ほとんどのも

のがコンバートされるはずで，このまま動作させてもタイトル表示までは動くはずです．

それ以上動作しないのは，BASE自身内部にリピート可能な1行入力ルーチンを持っていてそれがハードをアクセスしているためです．

FM-8のキー入力ルーチンはリピート付きなので，この部分はGETLにパッチすればOKです．具体的なアドレスは，調べればすぐ解ると思います．ここまでくれば，プリンタ関係とロードを除いてすべて動作します．

プリンタその他も数箇所のパッチで動作します．リストも公開されているので容易にできるでしょう．

私がパッチしたものをダンプ・リストに示します．前に述べたとおり，オブジェクト・レベルでのパッチなので，かなり無駄がありますが一応動作します．

使用方法はオリジナルBASEとほぼ同じです．各コマンドおよびエディタのコマンドを**表5，6**に示します．

BASEは，非常に優秀な開発ツールです．リストを公開していただいたキャリーラボさんに本当に感謝します！

エピローグ

グラフィックス関係をまったくサポートしていないので，なぁーんだと思われた方も多いと思いますが，これは，約2Kバイトの中へ納めねばならなかったからで，次のバージョンでは，それらをすべて含んだものをと考えていますがいつになることやら….

何はともあれ以上のプログラムを利用すれば，カセット・ベースでZ80カードが使えるのです．カードをお持ちの方は早速使ってもらいたいと思います．お持ちでない方は，早速お店へ買いに行きましょう!!

最後に，原稿の校正，その他に協力してくれた，渡辺昌彦氏に誌上を借りて，お礼申し上げます．

◘参考文献

1) I/O'79年8月号“モニタを解剖する”
2) I/O'81年2月号“Z-80ソフトカード用ディスアセンブラ”
3) マニュアル“システム解説”富士通

表3　MZ-80 Tope Loader

ラベル名	アドレス	処理内容
MOTOFF	7080	モーターを止める
MOTON	7083	モーターをスタート
BIOS	708F	F-BASICのBIOSをコール
B1 LOAD	709D	第1ブロック・ロード
B2 LOAD	70A7	第2ブロック・ロード
HEDSKP	70CE	リーダー音待ち合わせ
CHECK	70E4	1ビット同期
BYTEIN	70FB	1バイト入力
HEX 2	711F	Dregを表示
ACC PRT	7137	Areg内ASCIIコードを表示
MSGOUT	7157	XからODがでるまでメッセージ・プリント
MSG	7162	Xから00がでるまでメーセージ・プリント

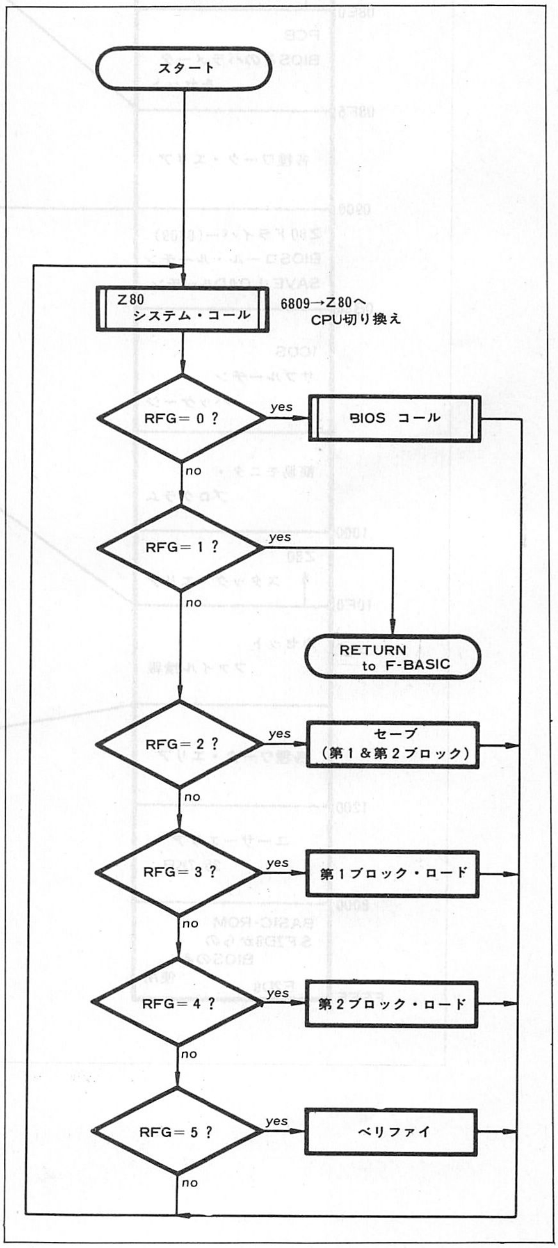
表4

アドレス	データ	変更データ
$7044	$20	$21
$7047, 48		ロード開始アドレス

表2　画面イニシャライズ・データ

アドレス	意味	データ
OBE0	背景色	00～07　黒～白
OBE1	横文字数	28H/50H　40字/80字
OBE2	縦文字数	14H/19H　20行/25行
OBE3	スクロール開始行	0～19H
OBE4	スクロール行数	0～19H
OBE5	ファンクション・キー表示	00/02　なし/あり
OBE7	カラースイッチ	00/FFH　カラー/白黒

フローチャート　1



メモリ・マップ

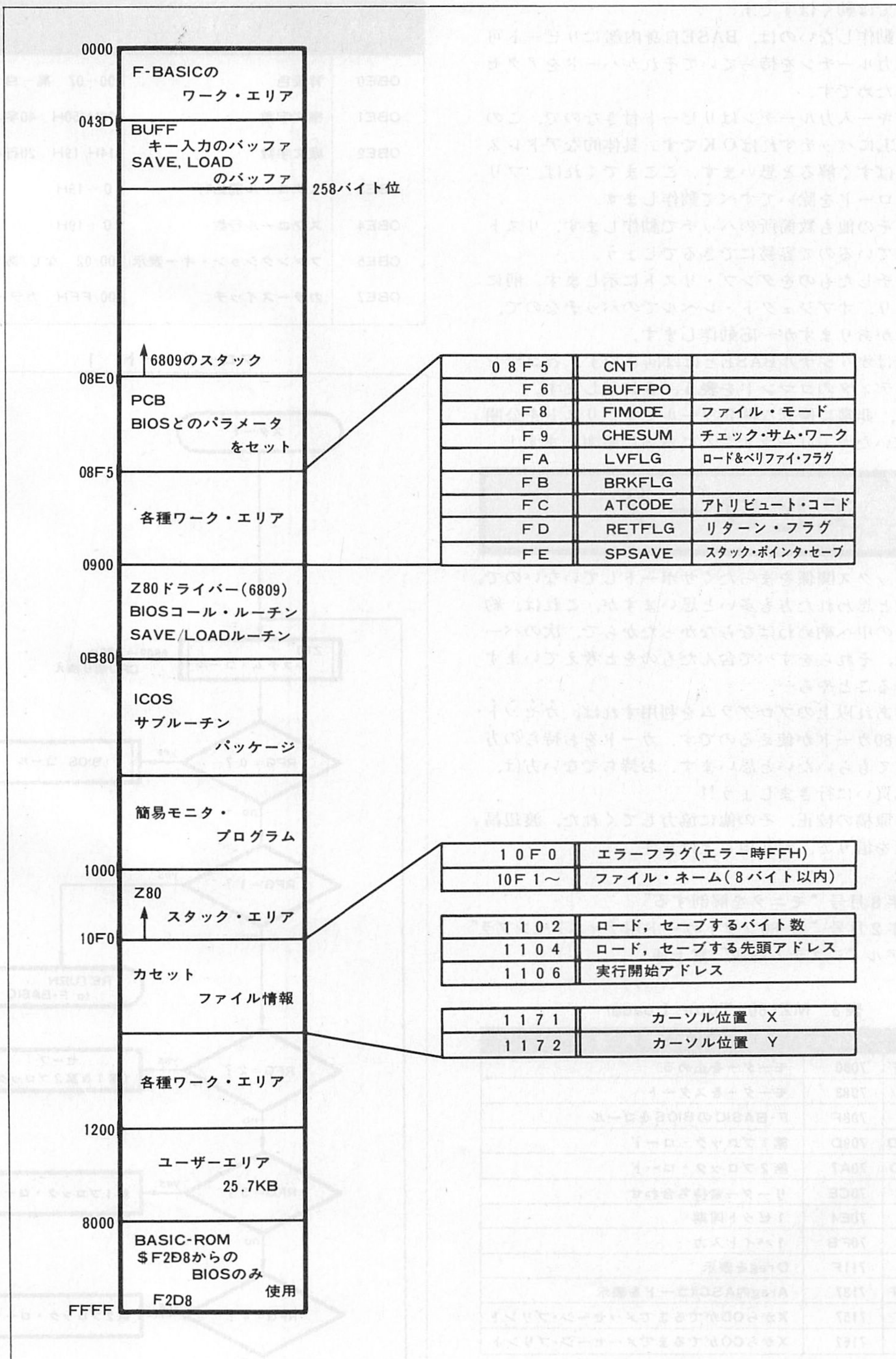

0000
F-BASICの
ワーク・エリア
043D
BUFF
キー入力のバッファ
SAVE, LOAD
のバッファ
258バイト位
6809のスタック
08E0
PCB
BIOSとのパラメータ
をセット
08F5
各種ワーク・エリア
0900
Z80ドライバー(6809)
BIOSコール・ルーチン
SAVE/LOADルーチン
0B80
ICOS
サブルーチン
パッケージ
簡易モニタ・
プログラム
1000
Z80
スタック・エリア
10F0
カセット
ファイル情報
各種ワーク・エリア
1200
ユーザーエリア
25.7KB
8000
BASIC-ROM
$F2D8からの
BIOSのみ
使用
F2D8
FFFF
08F5 CNT
F6 BUFFPO
F8 FIMODE ファイル・モード
F9 CHESUM チェック・サム・ワーク
FA LVFLG ロード&ベリファイ・フラグ
FB BRKFLG
FC ATCODE アトリビュート・コード
FD RETFLG リターン・フラグ
FE SPSAVE スタック・ポインタ・セーブ
10F0 エラーフラグ(エラー時FFH)
10F1～ ファイル・ネーム(8バイト以内)
1102 ロード, セーブするバイト数
1104 ロード, セーブする先頭アドレス
1106 実行開始アドレス
1171 カーソル位置 X
1172 カーソル位置 Y

表5 BASEメイン・コマンド表

コマンド	機能および使用法
A	BASE(アセンブル)する. A RET LABEL TABLE FROM? 5000 RET ラベル・テーブルの先頭アドレスを16進4桁で入力
A/	BASE＆リスト 使用法はAコマンドと同様 BASE完了後, BASEリストを出力
E	エディタをコールする エディタから抜けてメイン・コマンド待ち状態に 戻るときは R RET
Mm	m番地からメモリ・ダンプ CTRL C でコマンド待ちになる
Jm	m番地にジャンプ
Km	m番地から16進データを入力 7000␣00␣CD0030 RET 7003␣FF␣C3 RET X RET でコマンド待ちになる. ／ RET で1番地手前に戻る
O	ラベルの値を出力 O RET ですべてのラベルの値を出力 Oラベル名 RET で指定された ラベルの値を出力
#	プリンタ出力のON/OFF
!	モニタへ戻る

表6 BASEエディタ・コマンド

コマンド	機能	LPの移動
T	LPの示す行からリスト	無
TA	全行のリスト	無
Tn	n行からリスト	有
P	LPを最終行に移動	有
Pn	LPをn行に移動	有
PB	LPを先頭行に移動	有
Nn	LPをn進める	有
D	LPの示す行を削除	無
Dn	n行目を削除	有
I	LPの示す行からインサートする	有
Wf	ソース・テキストのセーブ	無
L	ソース・テキストのロード(テキスト先頭番地から)	無
Xm	テキスト・ポインタにmをセットする	有
.	テキストの先頭番地(テキスト・ポインタ)を表示	無
・	テキストの最終番地を表示	無
/	LPを表示	無
Jm	m番地にダイレクトでジャンプ	無
&	NEW	有
E	スクリーン・エディタを起動する	有
R	メインのコマンド待ちに戻る	無
Ss	sという文字列をサーチし, あればLPをその行にセットする	有
#	プリンタ出力のON/OFF	無
DPn	LPの示す行からn行削除	無

フローチャート2 MZ-80 Tape Loader

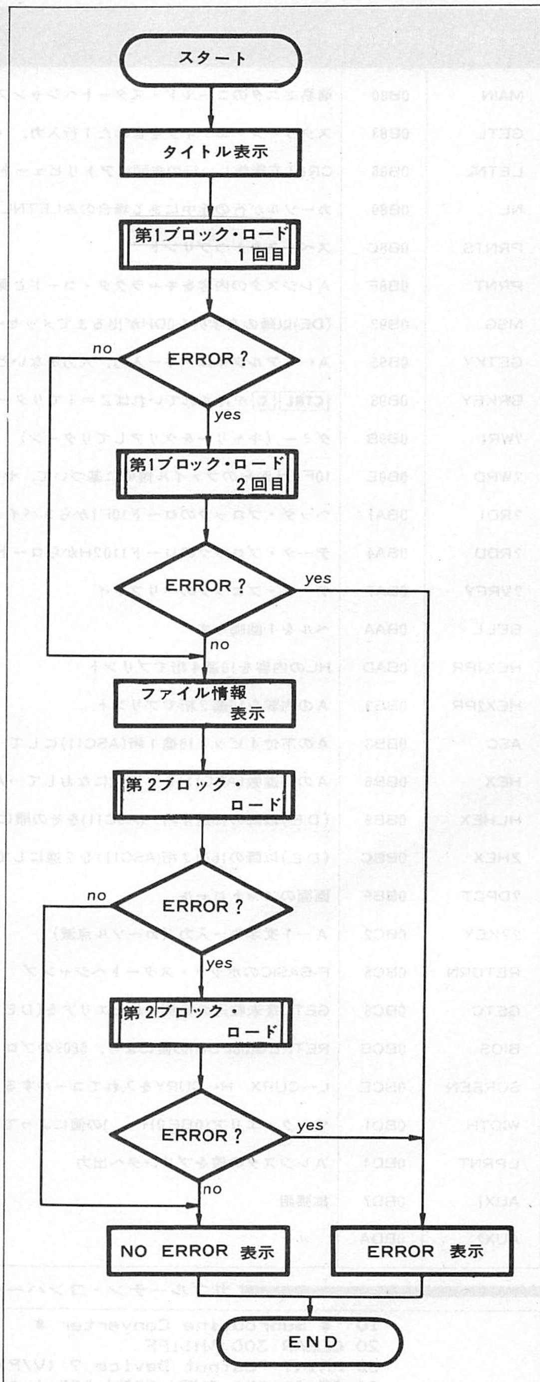


表1　サブルーチン仕様書

ラベル名	アドレス	機　　能	MZ^{2}のアドレス
MAIN	0B80	簡易モニタのコールド・スタートへジャンプ	0000
GETL	0B83	スクリーン・エディタを含めた1行入力．（DE）以降に転送	0003
LETNL	0B86	CR&LF後新しい行の先頭にアトリビュートを設定．アトリビュートは，ATCODEの値	0006
NL	0B89	カーソルが行の途中にある場合のみLETNLを実行	0009
PRNTS	0B8C	スペースを1つプリント	000C
PRNT	0B8F	Aレジスタの内容をキャラクタ・コードと見て，対応する文字をプリント	0012
MSG	0B92	(DE)以降の文字列を0DHが出るまでメッセージ・プリント	0015
GETKY	0B95	A←リアルタイム・キー入力．入力がないときは，A＝0	001B
BRKEY	0B98	CTRL C が押されていればZ＝1でリターン	001E
?WR1	0B9B	ダミー（キャリーをクリアしてリターン）	0021
?WRD	0B9E	10F1Hからのファイル情報に基づいて，セーブ(F-BASICと同じフォーマット)	0024
?RD1	0BA1	ヘッダ・ブロックのロード10F1から8バイトにファイル名がストアされます．	0027
?RDD	0BA4	データ・ブロックのロード1102Hからロードしたバイト数その他がストアされる．	002A
?VRFY	0BA7	データ・ブロックのベリファイ　　〃	002D
BELL	0BAA	ベルを1回鳴らす	003E
HEX4PR	0BAD	HLの内容を16進4桁でプリント	039A
HEX2PR	0BB0	Aの内容を16進2桁でプリント	03C3
ASC	0BB3	Aの下位4ビット16進1桁(ASC11)にして→A	03DA
HEX	0BB6	Aの16進数(ASC11)を2進数になおして→A	03F9
HLHEX	0BB9	（DE）以降の16進4桁　(ASC11)をその順に2進にして→HL	0410
ZHEX	0BBC	（DE）以降の16進2桁(ASC11)を2進にして→A	041F
?DPCT	0BBF	画面のコントロール	0DDC
??KEY	0BC2	A→1文字キー入力（カーソル点滅）	09B3
RETURN	0BC5	F-BASICのホット・スタートへジャンプ	
GETC	0BC8	GETL後未転送の変更されたエリアを(DE)以降に転送	
BIOS	0BCB	RETFLG(08FDH)の値により，6809のプログラムをコール（BIOSその他）	
SCREEN	0BCE	L←CURX　H←CURYを入れてコールするとその座標のキャラクタ・コードをAに入れてリターン	
WIDTH	0BD1	ワーク・エリア(0BE0H～　)の値によって画面モード・セット	
LPRNT	0BD4	Aレジスタの値をプリンタへ出力	
AUX1	0BD7	拡張用	
AUX2	0BDA	〃	

サブルーチン・コンバータ　プログラム(BASIC)リスト

```
10 '* Subroutine Converter *
20 CLEAR 300,&H11FF
30 PRINT "Output Device ? (V/P)":A$=INPUT$(1)
40 IF A$="P" THEN OPEN "O",1,"LPTO:" ELSE OPEN "O",1,"SCRN:"
50 INPUT "Start Address ";A$:I=VAL("&H"+A$)
60 INPUT "End Address    ";A$:E=VAL("&H"+A$)
70 A=PEEK(I)
80 IF A<>&HC3 AND A<>&HCD GOTO 160
90 C=PEEK(I+1)+PEEK(I+2)*256:IF C>&H1000 THEN 160
100 G$=RIGHT$("0000",4-LEN(HEX$(C)))+HEX$(C)
110 PRINT#1,HEX$(I)+"-"+HEX$(A)+G$+">";:RESTORE 170
120 READ A$,B$,LA$:IF A$="0" THEN PRINT#1,"NNNNNN ";:GOTO 160
130 A=VAL("&H"+A$):IF A<>C THEN 120
140 B=VAL("&H"+B$):POKE I+1,(B MOD 256):POKE I+2,INT(B/256):I=I+2
150 PRINT#1,LA$+" ";
160 I=I+1:IF I<=E THEN 70 ELSE PRINT#1:END
```

サブルーチン・コンバータ　プログラム(BASIC)リスト

```
170 DATA 3,B83,"GETL   ",6,B86,"LETNL ",9,B89,"NL     ",C,B8C,"PRNTS "
180 DATA F,BD7,"AUX1   ",12,B8F,"PRNT   ",15,B92,"MSG    ",18,B92,"MSG    "
190 DATA 1B,B95,"GETKY  ",1E,B98,"BRKEY  ",21,B9B,"?WR1   ",24,B9E,"?WRD   "
200 DATA 27,BA1,"?RD1   ",2A,BA4,"?RDD   ",2D,BA7,"?VRFY  ",30,BD7,"AUX1   "
210 DATA 33,BD7,"AUX1   ",3B,BD7,"AUX1   ",3E,BAA,"BELL   ",41,BD7,"AUX1   "
220 DATA 44,BD7,"AUX1   ",47,BD7,"AUX1   ",3BA,BAD,"HEX4PR",3C3,BB0,"HEX2PR"
230 DATA 3DA,BB3,"ASC    ",3F9,BB6,"HEX    ",410,BB9,"HLHEX ",41F,BBC,"2HEX  "
240 DATA DDC,BBF,"?DPCT ",9B3,BC2,"??KEY ",970,B8F,"PRNT   "
250 DATA BB9,BD7,"AUX1   ",BCE,BD7,"AUX1   ",DA6,BD7,"AUX1   ",DB5,B8F,"PRNT   "
260 DATA 0,0,END
```

MZ Tape Loader アセンブル・リスト

```
                        NAM   LOADER
                        OPT   M
                        OPT   NOP
                        OPT   NOG
                        OPT   S
                  ******************************
                  * MZ-80 Tape Loader          *
                  *    for FM-8 Version 1.0    *
                  *  Copyright (C)  HIRO Soft  *
                  * Written by T.Kasasaku      *
                  ******************************
7000                    ORG   $7000
                  *
7000 308D 0166 BEGIN    LEAX  MSG1,PCR
7004 17   015B          LBSR  MSG
7007 17   0079          LBSR  MOTON
700A 86   28            LDA   #40
700C A78D 01B9          STA   CNT,PCR
7010 17   00BB          LBSR  HEDSKP
7013 17   0087          LBSR  B1LOAD
7016 27   05            BEQ   LOAD1
7018 17   0082          LBSR  B1LOAD
701B 26   7A            BNE   ERR1
701D 308D 0171 LOAD1    LEAX  MSG3,PCR
7021 17   013E          LBSR  MSG
7024 308D 01A7          LEAX  BUFF+1,PCR
7028 17   012C          LBSR  MSGOUT
702B 308D 017B          LEAX  MSG6,PCR
702F 17   0130          LBSR  MSG
7032 308D 01AA          LEAX  BUFF+18,PCR
7036 EC   81            LDD   ,X++
7038 1E   89            EXG   A,B
703A 1F   02            TFR   D,Y
703C 34   20            PSHS  Y
703E EC   84            LDD   ,X
7040 1E   89            EXG   A,B
7042 1F   03            TFR   D,U
7044 20   03            BRA   LOADX
7046 CE   2000          LDU   #$2000
7049 17   00D3 LOADX    LBSR  HEX2
704C 86   2D            LDA   #'-
704E 17   00E6          LBSR  ACCPRT
7051 EC   81            LDD   ,X++
7053 1E   89            EXG   A,B
7055 E3   E1            ADDD  ,S++
7057 83   0001          SUBD  #1
705A 17   00C2          LBSR  HEX2
705D 86   2C            LDA   #',
705F 17   00D5          LBSR  ACCPRT
7062 EC   84            LDD   ,X
7064 1E   89            EXG   A,B
7066 17   00B6          LBSR  HEX2
7069 86   14            LDA   #20
706B A78D 015A          STA   CNT,PCR
706F 8D   5D            BSR   HEDSKP
7071 8D   34            BSR   B2LOAD
7073 27   04            BEQ   LOAD2
7075 8D   30            BSR   B2LOAD
7077 26   1E            BNE   ERR1
7079 308D 011F LOAD2    LEAX  MSG5,PCR
707D 17   00E2 LOAD3    LBSR  MSG
                  *
7080 4F        MOTOFF   CLRA
7081 20   02            BRA   MOTOR
7083 86   FF   MOTON    LDA   #$FF
7085 A78D 0139 MOTOR    STA   RCB+2,PCR
7089 86   01            LDA   #1
708B A78D 0131          STA   RCB,PCR
708F 308D 012D BIOS     LEAX  RCB,PCR
7093 6E9F FBFA          JMP   [$FBFA]
                  *
7097 308D 00E3 ERR1     LEAX  MSG2,PCR
709B 20   E0            BRA   LOAD3
                  *
709D 308D 012D B1LOAD   LEAX  BUFF,PCR
70A1 108E 0080          LDY   #128
70A5 20   02            BRA   BLLOAD
70A7 1F   31   B2LOAD   TFR   U,X
70A9 CC   0000 BLLOAD   LDD   #0
70AC ED8D 011A          STD   CHESUM,PCR
70B0 8D   49   BLOAD1   BSR   BYTEIN
70B2 A7   80            STA   ,X+
70B4 31   3F            LEAY  -1,Y
70B6 26   F8            BNE   BLOAD1
70B8 EC8D 010E          LDD   CHESUM,PCR
70BC ED8D 010C          STD   CHSUM1,PCR
70C0 8D   39            BSR   BYTEIN
70C2 1F   89            TFR   A,B
70C4 8D   35            BSR   BYTEIN
70C6 1E   89            EXG   A,B
70C8 10A38D00FF         CMPD  CHSUM1,PCR
70CD 39                 RTS
                  *
70CE E68D 00F7 HEDSKP   LDB   CNT,PCR
70D2 8D   10   SKP1     BSR   CHECK
70D4 24   F8            BCC   HEDSKP
70D6 5A                 DECB
70D7 26   F9            BNE   SKP1
70D9 E68D 00EC          LDB   CNT,PCR
70DD 8D   05   SKP2     BSR   CHECK
70DF 25   ED            BCS   HEDSKP
70E1 5A                 DECB
70E2 26   F9            BNE   SKP2
                  *
70E4 78   FD02 CHECK    ASL   $FD02
70E7 25   FB            BCS   CHECK
70E9 78   FD02 CHECK1   ASL   $FD02
70EC 24   FB            BCC   CHECK1
70EE 34   02            PSHS  A
70F0 86   4B            LDA   #75
70F2 4A        CHECK2   DECA
70F3 26   FD            BNE   CHECK2
70F5 35   02            PULS  A
70F7 78   FD02          ASL   $FD02
70FA 39                 RTS
                  *
70FB 34   04   BYTEIN   PSHS  B
70FD 8D   E5   BYTIN1   BSR   CHECK
70FF 24   FC            BCC   BYTIN1
7101 C6   08            LDB   #8
7103 8D   DF   BYTIN2   BSR   CHECK
7105 24   12            BCC   BYTIN3
7107 49                 ROLA
7108 34   06            PSHS  A,B
710A EC8D 00BC          LDD   CHESUM,PCR
710E C3   0001          ADDD  #1
7111 ED8D 00B5          STD   CHESUM,PCR
7115 35   06            PULS  A,B
7117 20   01            BRA   BYTIN4
7119 49        BYTIN3   ROLA
711A 5A        BYTIN4   DECB
711B 26   E6            BNE   BYTIN2
711D 35   84            PULS  B,PC
                  *
711F 8D   02   HEX2     BSR   HEXPRT
7121 1F   98            TFR   B,A
                  *
7123 34   02   HEXPRT   PSHS  A
7125 47                 ASRA
7126 47                 ASRA
7127 47                 ASRA
7128 47                 ASRA
7129 8D   02            BSR   HPUT
712B 35   02            PULS  A
                  *
712D 84   0F   HPUT     ANDA  #$0F
712F 81   0A            CMPA  #10
7131 25   02            BCS   HPUT1
7133 8B   07            ADDA  #7
7135 8B   30   HPUT1    ADDA  #$30
                  *
7137 34   10   ACCPRT   PSHS  X
7139 A78D 008B          STA   RCB+8,PCR
713D 86   14            LDA   #$14
713F A78D 007D          STA   RCB,PCR
7143 308D 0081          LEAX  RCB+8,PCR
7147 AF8D 0077          STX   RCB+2,PCR
714B 8E   0001          LDX   #1
714E AF8D 0072          STX   RCB+4,PCR
7152 17   FF3A          LBSR  BIOS
7155 35   90            PULS  X,PC
                  *
7157 A6   80   MSGOUT   LDA   ,X+
7159 81   0D            CMPA  #13
715B 27   04            BEQ   MGOUT1
715D 8D   D8            BSR   ACCPRT
```

MZTAPE Loader アセンブル・リスト

```
715F 20    F6           BRA   MSGOUT
7161 39          MGOUT1 RTS
                 *
7162 A6    80    MSG    LDA   ,X+
7164 27    FB           BEQ   MGOUT1
7166 8D    CF           BSR   ACCPRT
7168 20    F8           BRA   MSG
                 *
716A 0D          MSG1   FCB   13,10
716C 53                 FCC   'Searching'
7175 20                 FCC   ' MZ Tape'
717D 00                 FCB   0
717E 0D          MSG2   FCB   13,10,7
7181 44                 FCC   'Device I/O'
718B 20                 FCC   ' Error'
7191 00                 FCB   0
7192 0D          MSG3   FCB   13,10
7194 46                 FCC   'Found: '
719B 00                 FCB   0
719C 20          MSG5   FCC   ' ---'
71A0 20                 FCC   ' No Errors'
71AA 0D          MSG6   FCB   13,10,0
                 *
71AD 45                 FCC   'END '
71B1 20                 FCC   ' I love HITOMI.'
                 *
71C0 0009        RCB    RMB   9
71C9 0001        CNT    RMB   1
71CA 0002        CHESUM RMB   2
71CC 0002        CHSUM1 RMB   2
71CE 0080        BUFF   RMB   128
                        END
```

Z80 ドライバー アセンブル・リスト

```
                                NAM   Z-80     DRIVER
                                OPT   NOG
                                OPT   NOP
                                OPT   M
           08                   SETDP $08
                 ******************************
                 * Z-80 DRIVER Version 1.0    *
                 *   BIOS <==> Z-80 System    *
                 * Copyright (C) 1982         *
                 *   By HIRO/ETF Laboratory   *
                 * Written by T.Kasasaku      *
                 ******************************
           043D  BUFF    EQU   $043D
           08E0  RCB     EQU   $08E0
           08F5  CNT     EQU   $08F5
           08F6  BUFFPO  EQU   $08F6
           08F8  FIMODE  EQU   $08F8
           08F9  CHESUM  EQU   $08F9
           08FA  LVFLG   EQU   $08FA
           08FB  AUX     EQU   $08FB
           08FC  ATCODE  EQU   $08FC
           08FD  RETFLG  EQU   $08FD
           08FE  SPSAVE  EQU   $08FE
           10F0  ERRFLG  EQU   $10F0
           10F1  FINAME  EQU   $10F1
           1102  BYTES   EQU   $1102
           1104  BEGADD  EQU   $1104
           1106  EXEADD  EQU   $1106
                 *
0900                     ORG   $900
                 *
                 * Z-80 Start Addr. Set
                 *
0900 10FF  08FE  BEGIN   STS   >SPSAVE
0904 10CE  08DF          LDS   #RCB-1
0908 34    08            PSHS  DP
090A 86    08            LDA   #$08
090C 1F    8B            TFR   A,DP
090E 86    C3            LDA   #$C3
0910 B7    0000          STA   $0000
0913 CC    800B          LDD   #$800B
0916 FD    0001          STD   $0001
0919 20    17            BRA   LOOP1
                 *
                 * MAIN Process
                 *
091B 96    FD    LOOP    LDA   RETFLG
091D 27    23            BEQ   LOOP2
091F 4A                  DECA
0920 1027  021D          LBEQ  RETURN
0924 4A                  DECA
0925 1027  0116          LBEQ  SAVE
0929 4A                  DECA
092A 27    26            BEQ   B1LOAD
092C 4A                  DECA
092D 27    54            BEQ   B2LOAD
092F 4A                  DECA
0930 27    4D            BEQ   VERIFY
                 *
                 * Jump to Z-80
                 *
0932 0F    FD    LOOP1   CLR   RETFLG
0934 10CE  08DF          LDS   #RCB-1
0938 1A    40            ORCC  #$40
093A 86    01            LDA   #1
093C B7    FD05          STA   $FD05
093F 12                  NOP
0940 20    D9            BRA   LOOP
                 *
                 * Call BIOS
                 *
0942 7F    0313  LOOP2   CLR   $0313
0945 17    00AA          LBSR  BIOS
0948 20    E8            BRA   LOOP1
                 *
                 * CMT Error
                 *
094A 73    10F0  ERR1    COM   ERRFLG
094D 17    009B          LBSR  MOTOFF
0950 20    E0            BRA   LOOP1
                 *
                 * Head Block Load
                 *
0952 17    0092  B1LOAD  LBSR  MOTON
0955 7F    10F0          CLR   ERRFLG
0958 17    009E  LOAD2   LBSR  BLOAD
095B 26    ED            BNE   ERR1
095D 8E    043D          LDX   #BUFF
0960 A6    81            LDA   ,X++
0962 26    F4            BNE   LOAD2
0964 A6    08            LDA   8,X
0966 81    02            CMPA  #2
0968 26    E0            BNE   ERR1
096A 108E  10F1          LDY   #FINAME
096E C6    08            LDB   #8
0970 A6    80    LOAD3   LDA   ,X+
0972 A7    A0            STA   ,Y+
0974 5A                  DECB
0975 26    F9            BNE   LOAD3
0977 86    0D            LDA   #13
0979 A7    A4            STA   ,Y
097B 8D    6E            BSR   MOTOFF
097D 20    B3    LOAD3A  BRA   LOOP1
                 *
                 * Data Block Verify
                 *
097F 0F    FA    VERIFY  CLR   LVFLG
0981 20    04            BRA   LOAD3B
                 *
                 * Data Block Load
                 *
0983 86    01    B2LOAD  LDA   #1
0985 97    FA            STA   LVFLG
0987 8D    5E    LOAD3B  BSR   MOTON
0989 8D    6E            BSR   BLOAD
098B 26    BD            BNE   ERR1
098D 8E    043D          LDX   #BUFF
0990 A6    80            LDA   ,X+
0992 81    01            CMPA  #1
0994 26    B4            BNE   ERR1
0996 30    02            LEAX  2,X
0998 EC    81            LDD   ,X++
099A 1F    02            TFR   D,Y
099C 1E    89            EXG   A,B
099E FD    1102          STD   BYTES
09A1 EC    84            LDD   ,X
09A3 1F    03            TFR   D,U
09A5 1E    89            EXG   A,B
09A7 FD    1104          STD   BEGADD
09AA 8E    0444          LDX   #BUFF+7
09AD F6    043E          LDB   BUFF+1
09B0 C0    05            SUBB  #5
09B2 A6    80    LOAD4   LDA   ,X+
09B4 0D    FA            TST   LVFLG
09B6 26    06            BNE   LOAD4A
09B8 A1    C0            CMPA  ,U+
09BA 26    8E            BNE   ERR1
09BC 20    02            BRA   LOAD4B
09BE A7    C0    LOAD4A  STA   ,U+
09C0 31    3F    LOAD4B  LEAY  -1,Y
09C2 27    10            BEQ   LOAD5
09C4 5A                  DECB
09C5 26    EB            BNE   LOAD4
09C7 8D    30            BSR   BLOAD
09C9 1026  FF7D          LBNE  ERR1
09CD 8E    043E          LDX   #BUFF+1
09D0 E6    80            LDB   ,X+
09D2 20    DE            BRA   LOAD4
09D4 63    80    LOAD5   COM   ,X+
09D6 1026  FF70          LBNE  ERR1
09DA 30    02            LEAX  2,X
09DC EC    84            LDD   ,X
09DE 1E    89            EXG   A,B
09E0 FD    1106          STD   EXEADD
09E3 8D    06            BSR   MOTOFF
09E5 20    96            BRA   LOAD3A
                 *
09E7 86    FF    MOTON   LDA   #$FF
09E9 20    01            BRA   MOTOR
09EB 4F          MOTOFF  CLRA
09EC 97    E2    MOTOR   STA   RCB+2
09EE 86    01            LDA   #1
```

Z80 ドライバー アセンブル・リスト

```
09F0 97     E0          STA    RCB
09F2 8E     08E0 BIOS   LDX    #RCB
09F5 6E9F   FBFA        JMP    [$FBFA]
                 *
                 * One-Block Load
                 *
09F9 86     03   BLOAD  LDA    #3
09FB 97     E0          STA    RCB
09FD 8E     043D        LDX    #BUFF
0A00 8D     2C   BLOAD1 BSR    CMTIN
0A02 81     01          CMPA   #1
0A04 26     FA          BNE    BLOAD1
0A06 8D     26          BSR    CMTIN
0A08 81     3C          CMPA   #$3C
0A0A 26     F4          BNE    BLOAD1
0A0C 8D     20          BSR    CMTIN
0A0E 97     F9          STA    CHESUM
0A10 A7     80          STA    ,X+
0A12 8D     1A   BLOAD2 BSR    CMTIN
0A14 A7     80          STA    ,X+
0A16 27     15          BEQ    BLOAD4
0A18 1F     89          TFR    A,B
0A1A 9B     F9          ADDA   CHESUM
0A1C 97     F9          STA    CHESUM
0A1E 8D     0E   BLOAD3 BSR    CMTIN
0A20 A7     80          STA    ,X+
0A22 9B     F9          ADDA   CHESUM
0A24 97     F9          STA    CHESUM
0A26 5A                 DECB
0A27 26     F5          BNE    BLOAD3
0A29 8D     03          BSR    CMTIN
0A2B 91     F9          CMPA   CHESUM
0A2D 39          BLOAD4 RTS
                 *
                 * 1 Byte Input
                 *
0A2E 34     10   CMTIN  PSHS   X
0A30 B6     FD04        LDA    $FD04
0A33 84     02          ANDA   #2
0A35 1027   FF11        LBEQ   ERR1
0A39 8D     B7          BSR    BIOS
0A3B 96     E2          LDA    RCB+2
0A3D 35     90          PULS   X,PC
                 *
                 * Save Routine
                 *
0A3F 8E     043D SAVE   LDX    #BUFF
0A42 9F     F6          STX    BUFFPO
0A44 C6     08          LDB    #8
0A46 0F     F5          CLR    CNT
0A48 8E     10F1        LDX    #FINAME
0A4B A6     80   SAVE1  LDA    ,X+
0A4D 17     009C        LBSR   SDRIVE
0A50 5A                 DECB
0A51 26     F8          BNE    SAVE1
0A53 FC     1104        LDD    BEGADD
0A56 1E     89          EXG    A,B
0A58 1F     03          TFR    D,U
0A5A FC     1102        LDD    BYTES
0A5D 1E     89          EXG    A,B
0A5F 1F     02          TFR    D,Y
0A61 8D     84          BSR    MOTON
0A63 17     00CE        LBSR   WAIT
0A66 0F     F8          CLR    FIMODE
0A68 8D     5E          BSR    SAVE9
0A6A 86     02          LDA    #2
0A6C 8D     7E          BSR    SDRIVE
0A6E C6     0B          LDB    #11
0A70 4F          SAVE5  CLRA
0A71 8D     79          BSR    SDRIVE
0A73 5A                 DECB
0A74 26     FA          BNE    SAVE5
0A76 17     0087        LBSR   BLSAVE
0A79 5F                 CLRB
0A7A 86     FF          LDA    #$FF
0A7C 8D     54   SAVE6  BSR    CMTOUT
0A7E 5A                 DECB
0A7F 26     FB          BNE    SAVE6
0A81 0C     F8          INC    FIMODE
0A83 4F                 CLRA
0A84 8D     66          BSR    SDRIVE
0A86 1F     20          TFR    Y,D
0A88 8D     5E          BSR    DSDRV
0A8A 1F     30          TFR    U,D
0A8C 8D     5A          BSR    DSDRV
0A8E A6     C0   SAVE7  LDA    ,U+
0A90 8D     5A          BSR    SDRIVE
0A92 31     3F          LEAY   -1,Y
0A94 26     F8          BNE    SAVE7
0A96 86     FF          LDA    #$FF
0A98 8D     52          BSR    SDRIVE
0A9A CC     0000        LDD    #0
0A9D 8D     49          BSR    DSDRV
0A9F FC     1106        LDD    EXEADD
0AA2 1E     89          EXG    A,B
0AA4 8D     42          BSR    DSDRV
0AA6 8E     043D        LDX    #BUFF
0AA9 9C     F6          CMPX   BUFFPO
0AAB 27     02          BEQ    SAVE8
0AAD 8D     51          BSR    BLSAVE
0AAF 86     01   SAVE8  LDA    #1
0AB1 8D     1F          BSR    CMTOUT
0AB3 86     3C          LDA    #$3C
0AB5 8D     1B          BSR    CMTOUT
0AB7 86     FF          LDA    #$FF
0AB9 8D     17          BSR    CMTOUT
0ABB 4F                 CLRA
0ABC 8D     14          BSR    CMTOUT
0ABE 8D     08          BSR    SAVE9
0AC0 8D     72          BSR    WAIT
0AC2 17     FF26        LBSR   MOTOFF
0AC5 16     FE6A        LBRA   LOOP1
                 *
0AC8 C6     0E   SAVE9  LDB    #14
0ACA 86     FF          LDA    #$FF
0ACC 8D     04   SAVE10 BSR    CMTOUT
0ACE 5A                 DECB
0ACF 26     FB          BNE    SAVE10
0AD1 39                 RTS
                 *
                 * 1 Byte Output
                 *
0AD2 34     12   CMTOUT PSHS   A,X
0AD4 97     E2          STA    RCB+2
0AD6 86     02          LDA    #2
0AD8 97     E0          STA    RCB
0ADA 17     FF15        LBSR   BIOS
0ADD B6     FD04        LDA    $FD04
0AE0 84     02          ANDA   #2
0AE2 1027   FE64        LBEQ   ERR1
0AE6 35     92          PULS   A,X,PC
                 *
0AE8 8D     02   DSDRV  BSR    SDRIVE
0AEA 1F     98          TFR    B,A
0AEC 34     10   SDRIVE PSHS   X
0AEE 9E     F6          LDX    BUFFPO
0AF0 A7     80          STA    ,X+
0AF2 9F     F6          STX    BUFFPO
0AF4 0C     F5          INC    CNT
0AF6 96     F5          LDA    CNT
0AF8 81     FF          CMPA   #$FF
0AFA 26     02          BNE    SDRIV1
0AFC 8D     02          BSR    BLSAVE
0AFE 35     90   SDRIV1 PULS   X,PC
                 *
0B00 34     10   BLSAVE PSHS   X
0B02 8E     043D        LDX    #BUFF
0B05 9F     F6          STX    BUFFPO
0B07 86     01          LDA    #1
0B09 8D     C7          BSR    CMTOUT
0B0B 86     3C          LDA    #$3C
0B0D 8D     C3          BSR    CMTOUT
0B0F 96     F8          LDA    FIMODE
0B11 8D     BF          BSR    CMTOUT
0B13 97     F9          STA    CHESUM
0B15 96     F5          LDA    CNT
0B17 0F     F5          CLR    CNT
0B19 8D     B7          BSR    CMTOUT
0B1B 1F     89          TFR    A,B
0B1D 9B     F9          ADDA   CHESUM
0B1F 97     F9          STA    CHESUM
0B21 A6     80   BSAVE1 LDA    ,X+
0B23 8D     AD          BSR    CMTOUT
0B25 9B     F9          ADDA   CHESUM
0B27 97     F9          STA    CHESUM
0B29 5A                 DECB
0B2A 26     F5          BNE    BSAVE1
0B2C 96     F9          LDA    CHESUM
0B2E 8D     A2          BSR    CMTOUT
0B30 8D     96          BSR    SAVE9
0B32 35     90          PULS   X,PC
                 *
0B34 86     0A   WAIT   LDA    #10
0B36 8E     0000 WAIT1  LDX    #0
0B39 30     1F   WAIT2  LEAX   -1,X
0B3B 26     FC          BNE    WAIT2
0B3D 4A                 DECA
0B3E 26     F6          BNE    WAIT1
0B40 39                 RTS
                 *
                 * Return to F-BASIC
                 *
0B41 35     08   RETURN PULS   DP
0B43 10FE   08FE        LDS    >SPSAVE
0B47 6E9F   FBFC        JMP    [$FBFC]
                 *
0B4B 43                 FCC    'Copyright (C) '
0B59 4A                 FCC    'JI1ETF JM1EQV '
0B67 4A                 FCC    'JM1FXM JN1XUL '
0B75 31                 FCC    '1982/04/12'
                 *
                        END
```

BASE FM-8 ダンプ・リスト

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
1200	C3	5E	2D	C3	B9	2D	01	25	34	00	34	A7	34	08	00	25	:8D
1210	34	00	60	01	60	04	00	01	60	00	00	00	00	20	34	00	:AE
1220	60	00	00	00	00	00	00	03	00	08	03	00	0E	00	00	00	:7C
1230	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
1240	40	12	03	40	01	00	00	CD	86	0B	18	03	CD	89	0B	3A	:AA

BASE FM-8 ダンプ・リスト

```
1250 45 12 B7 C8 3E 0D C3 48 30 FE 20 30 06 F5 CD 8F :01
1260 0B F1 C9 F5 CD D7 0B CD 8F 0B F1 C9 3E 20 18 10 :10
1270 C3 C2 0B C3 95 0B 1A 13 FE 0D C8 CD 80 12 18 F6 :60
1280 F5 3A 45 12 B7 28 05 F1 F5 CD 48 30 F1 CD 63 12 :C8
1290 C9 7C CD 96 12 7D F5 E6 F0 0F 0F 0F 0F CD A3 12 :C0
12A0 F1 E6 0F B7 27 C6 F0 CE 40 18 D5 21 00 00 3E 10 :E4
12B0 EB 29 EB ED 6A B7 ED 42 30 03 09 18 01 13 3D C2 :A3
12C0 B0 12 C9 21 A8 11 06 06 36 20 2B 10 FB 21 A8 11 :D7
12D0 36 0D E5 01 0A 00 CD AB 12 7D C6 30 E1 2B 77 7A :2D
12E0 B3 C8 18 EE CD 4C 12 C5 E5 D5 CD C3 12 11 A4 11 :93
12F0 CD 76 12 D1 E1 C1 C9 D5 29 54 5D 29 29 19 D1 C9 :45
------------------------------------------------------------
Sum: AA 57 FF B1 74 60 6E 50 82 E6 78 14 EB FB 51 4F :BD


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1300 21 00 00 1A FE 20 20 03 13 18 F8 1A 13 FE 30 D8 :D2
1310 FE 3A D0 CD F7 12 D6 30 06 00 4F 09 18 ED ED 5B :8F
1320 0F 12 1A B7 28 08 FE 0D 28 03 13 18 F5 13 ED 53 :CB
1330 0F 12 C9 ED 5B 0F 12 ED 53 07 12 C9 ED 5B 07 12 :D6
1340 ED 53 0F 12 C9 ED 5B 07 12 1A 13 B7 20 06 1B ED :9D
1350 53 07 12 C9 FE 0D C2 49 13 ED 53 07 12 C9 ED 5B :C8
1360 09 12 ED 53 0F 12 44 4D 0B 78 B1 C8 CD 1E 13 0B :12
1370 1A B7 C8 18 F4 21 FF FF 22 0D 12 2A 09 12 22 0F :7B
1380 12 18 03 CD D8 13 2A 0F 12 7E B7 C8 CD A4 13 CD :7E
1390 95 0B 28 F2 CD 70 12 FE 64 C8 FE 58 C8 FE 45 CA :5E
13A0 62 15 18 E2 ED 5B 0D 12 13 ED 53 0D 12 13 CD E4 :0E
13B0 12 CD 6C 12 ED 5B 0F 12 1A B7 C8 FE 20 20 10 3A :E7
13C0 45 12 FE 01 20 09 06 04 3E 20 CD 80 12 10 F9 CD :1C
13D0 D7 0B CD 76 12 C3 1E 13 ED 5B 09 12 2A 07 12 22 :F3
13E0 0F 12 B7 ED 52 44 4D 62 6B 11 FF FF 78 B1 28 08 :DD
13F0 3E 0D ED B1 13 EA F2 13 ED 53 0D 12 C9 11 A5 11 :DA
------------------------------------------------------------
Sum: 24 C2 A7 99 58 A9 21 86 0C 77 47 82 59 06 5B B7 :8B


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1400 CD 00 13 CD 5E 13 2A 0F 12 22 07 12 C9 CD FD 13 :4A
1410 C3 83 13 2A 09 12 AF C5 01 00 00 ED B1 C1 2B 22 :BF
1420 0B 12 C9 1A FE 0D CA 83 13 FE 41 CA 75 13 18 DD :F1
1430 CD B9 0B D8 22 09 12 CD 13 14 ED 5B 09 12 ED 53 :3D
1440 07 12 ED 53 0F 12 C9 2A 09 12 36 00 18 EC CD D8 :67
1450 13 2A 0D 12 23 23 EB C3 E4 12 2A 09 12 C3 91 12 :F1
1460 CD 13 14 C3 91 12 CD B9 0B D8 D1 E9 1A FE 42 28 :FF
1470 C9 18 8A CD 00 13 44 4D CD 45 13 B7 C8 0B 78 B1 :B4
1480 C2 78 14 C9 2A 07 12 CD 3C 13 CD 1E 13 ED 5B 0F :CB
1490 12 1A 13 77 23 B7 C2 91 14 C9 13 CD 00 13 44 4D :44
14A0 78 B1 C8 0B CD 84 14 18 F7 1A FE 50 28 EC FE 0D :F7
14B0 28 D2 CD 00 13 CD 5E 13 2A 0F 12 22 07 12 18 C4 :7A
14C0 D5 01 01 00 2A 07 12 7E B7 28 04 03 23 18 F8 E5 :96
14D0 1A FE 0D 28 04 13 23 18 F7 23 7C FE C0 30 1A 36 :73
14E0 DD 7E FE DD 20 13 EB E1 ED B8 2A 07 12 D1 1A 77 :7F
14F0 FE 0D CA 45 13 23 13 18 F5 CD 26 16 2A 4D 45 4D :82
------------------------------------------------------------
Sum: 56 54 24 73 D8 F4 F3 2F FF 4A 39 48 65 CF 6B 34 :CC


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1500 4F 52 59 20 46 55 4C 4C 00 E1 D1 C9 11 A3 11 CD :5A
1510 10 31 1A FE 1B C8 CD C0 14 18 F1 2A 07 12 22 0F :5A
1520 12 EB 21 A5 11 42 4B 0A BE 28 14 7E FE 0D 20 02 :10
1530 B7 C9 1A FE 0D 20 02 37 C9 21 A5 11 13 18 E6 7E :2D
1540 FE 0D 20 02 B7 C9 03 23 18 DD 13 1A FE 0D C8 CD :95
1550 1B 15 30 08 CD 45 13 1A B7 C8 18 F3 CD D8 13 C3 :AC
1560 A4 13 11 A3 11 CD 10 31 CD 00 13 7C B5 C8 D5 CD :05
1570 5E 13 2A 0F 12 22 07 12 CD 84 14 D1 CD C0 14 18 :E6
1580 E1 11 81 15 D5 CD 4C 12 3E 23 CD 59 12 11 A3 11 :E6
1590 CD 10 31 1A FE 23 20 D0 13 1A 13 FE 26 CA 47 14 :C2
15A0 FE 53 28 A6 FE 54 CA 23 14 FE 2C CA 5A 14 FE 2E :00
15B0 CA 60 14 FE 2F CA 4E 14 FE 50 CA 6C 14 FE 4A CA :41
15C0 66 14 FE 58 CA 30 14 FE 44 CA A9 14 FE 49 CA 0C :C4
15D0 15 FE 23 CA 18 30 FE 4E CA 73 14 FE 4C CA 81 2F :A9
15E0 FE 45 CA 62 15 00 00 00 00 FE 57 20 04 AF C3 25 :94
15F0 2F FE 52 20 02 D1 C9 FE 0D C8 CD 26 16 43 4F 4D :F6
------------------------------------------------------------
Sum: 61 A8 64 F4 1F BB F2 30 82 F9 84 C1 80 39 8C 9B :FD


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1600 4D 41 4E 44 20 45 52 52 0D 00 C9 1A FE 41 20 09 :81
1610 CD 13 14 22 03 C0 C3 05 C0 FE 57 20 06 13 3E 01 :2E
1620 C3 25 2F C3 05 C0 E3 7E 23 E3 B7 C8 CD 59 12 18 :D5
1630 F5 E3 ED 5B 0F 12 EB 1A 13 B7 28 21 FE 0D 28 0E :9A
1640 BE 23 28 F3 1A 13 B7 20 FB EB AF 3C E3 C9 BE 28 :63
1650 0B 7E FE 20 28 06 FE 3A 28 02 18 E8 13 EB ED 53 :75
1660 0F 12 E3 C9 E5 2A 0F 12 23 22 0F 12 E1 C9 E5 21 :13
1670 00 00 ED 5B 0F 12 1A FE 30 38 13 FE 3A 30 0F D6 :49
1680 30 29 44 4D 29 29 09 4F 06 00 09 13 18 E8 ED 53 :F6
1690 0F 12 EB E1 C9 E5 21 00 00 ED 5B 0F 12 13 1A FE :50
16A0 30 38 EB FE 47 30 E7 FE 3A 38 06 FE 41 30 06 18 :B2
16B0 DD D6 30 18 02 D6 37 29 29 29 29 4F 06 00 09 18 :24
16C0 DC ED 5B 0F 12 13 1A 13 13 ED 53 0F 12 5F 16 00 :6E
16D0 C9 01 00 00 2A 11 12 ED 5B 0F 12 7E B7 CA C1 17 :57
16E0 1A BE 20 08 7E FE 0D C8 13 23 18 F4 7E FE 0D 28 :44
16F0 0A 23 7E FE 0D 20 FA 23 03 18 DC 1A FE 20 C8 FE :E8
------------------------------------------------------------
Sum: BF 27 B7 14 6F 82 3C BA 66 64 D4 61 96 D9 F9 60 :5F


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1700 3A C8 FE 2B C8 FE 2D C8 FE 29 C8 FE 2F C8 18 E7 :C9
1710 60 69 29 ED 4B 13 12 09 5E 23 56 C9 E5 CD D1 16 :91
1720 ED 53 0F 12 CD 10 17 E1 C9 ED 5B 0F 12 1A FE 24 :A4
1730 CA 95 16 FE 22 28 8A FE 30 38 07 FE 3A 30 03 C3 :E2
1740 6E 16 1A FE 3A 20 04 11 00 00 C9 FE 20 20 04 11 :27
1750 00 00 C9 FE 29 20 04 11 00 00 C9 18 BF CD 29 17 :D2
1760 D5 ED 5B 0F 12 1A FE 2F 28 2F FE 2B 28 1E FE 2D :76
1770 28 35 FE 3A 20 02 D1 C9 FE 20 20 02 D1 C9 FE 29 :52
1780 20 02 D1 C9 FE 0D 20 02 D1 C9 18 2D CD 64 16 CD :DC
1790 29 17 62 6B D1 19 EB 18 C7 CD 64 16 CD 29 17 42 :57
17A0 4B D1 CD AB 12 18 B9 CD 64 16 CD 29 17 7B 2F 5F :D4
17B0 7A 2F 57 13 E1 19 EB 18 A7 1E 31 18 0E 1E 32 18 :94
17C0 0A 1E 33 18 06 1E 34 18 02 1E 35 CD AA 0B CD 47 :CE
17D0 12 CD 26 16 2A 45 52 52 20 00 7B CD 59 12 CD 4C :1A
17E0 12 ED 5B 07 12 CD D8 13 CD A4 13 C3 B9 2D E5 2A :67
17F0 0F 12 7E FE 0D 28 0E FE 20 20 03 23 18 F4 FE 3A :88
------------------------------------------------------------
Sum: 07 54 11 92 A8 54 D2 44 2D 6C 70 1B CB 17 1E DF :13


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1800 20 03 23 18 ED 22 0F 12 E1 C9 E5 2A 0F 12 7E 23 :09
1810 FE 0D 20 FA 22 0F 12 E1 C9 ED 5B 0F 12 D5 E5 21 :56
1820 25 12 36 00 E1 CD 31 16 28 49 58 2B 29 00 28 53 :FA
1830 D1 ED 53 0F 12 D5 CD 31 16 28 49 59 2B 29 00 28 :61
1840 42 D1 ED 53 0F 12 CD FE 18 B7 C0 CD 31 16 28 49 :53
1850 58 2B 00 20 04 0E 11 18 12 CD 31 16 28 49 59 2B :F9
1860 00 20 04 0E 14 18 04 0E 00 AF C9 C5 CD 5D 17 7B :69
1870 32 25 12 CD 31 16 29 0D 00 28 05 C1 0E 00 79 C9 :F1
1880 C1 79 C9 D1 ED 53 0F 12 CD FE 18 C9 ED 5B 0F 12 :4A
1890 D5 E5 21 25 12 36 00 E1 CD 31 16 28 49 58 2B 29 :5A
18A0 00 28 52 D1 ED 53 0F 12 D5 CD 31 16 28 49 59 2B :8A
18B0 29 00 28 41 D1 ED 53 0F 12 CD 5E 19 B7 C0 CD 31 :7D
18C0 16 28 49 58 2B 00 20 04 0E 11 18 12 CD 31 16 28 :B3
18D0 49 59 2B 00 20 04 0E 14 18 04 0E 00 AF C9 C5 CD :47
18E0 5D 17 7B 32 25 12 CD 31 16 29 00 28 05 C1 0E 00 :91
18F0 79 C9 C1 79 C9 D1 ED 53 0F 12 CD 5E 19 C9 21 AB :50
------------------------------------------------------------
Sum: D4 37 E3 7A 50 D1 83 1B DE 9B 50 DE 58 0C 06 AE :E6


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1900 1D 18 08 21 E7 1D 18 03 21 81 1A 0E 01 ED 5B 0F :9F
1910 12 7E B7 28 26 1A BE 20 08 FE 2F 28 1E 13 23 18 :56
1920 F4 7E FE 2F 20 0B 1A FE 3A C8 FE 0D C8 FE 20 C8 :9D
1930 7E 23 0C FE 2F 28 D6 7E 23 18 F8 0E 00 C9 E5 CD :12
1940 08 19 79 B7 28 04 ED 53 0F 12 E1 C9 E5 CD 03 19 :56
1950 18 F0 E5 CD 8C 18 18 EA E5 CD 19 18 18 E4 21 AB :0B
1960 1D 18 03 21 EF 1C 0E 01 ED 5B 0F 12 7E B7 28 CB :04
1970 1A BE 20 08 FE 2F 28 C3 13 23 18 F4 7E FE 2F C8 :CD
1980 0C 7E 23 FE 2F 28 E1 18 F8 E5 CD 63 19 18 B3 AF :9B
1990 32 28 12 32 29 12 32 2A 12 32 2B 12 2A 0F 12 E5 :E6
19A0 CD 31 16 2B 00 20 09 E5 21 28 12 36 01 E1 18 0E :E6
19B0 CD 31 16 2D 00 20 07 E5 21 28 12 36 02 E1 CD 52 :E0
19C0 19 B7 28 55 32 2A 12 CD 31 16 2B 00 20 09 E5 21 :29
19D0 29 12 36 01 E1 18 0E CD 31 16 2D 00 20 07 E5 21 :E7
19E0 29 12 36 02 E1 3A 1F 12 B7 28 08 CD 31 16 29 3D :20
19F0 00 18 06 CD 31 16 29 0D 00 20 1E 3A 2A 12 FE 07 :21
------------------------------------------------------------
Sum: 3B 11 45 D0 7A DD 8C 65 DF 97 FA 20 C1 4E 99 8D :6E


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1A00 30 17 FE 05 38 0D FE 05 3E DD 28 02 3E FD 32 2B :6F
1A10 12 3E 03 32 2A 12 E1 AF C9 E1 22 0F 12 F6 05 C9 :02
1A20 3A 28 12 B7 28 12 3D CD 7D 1A C6 03 4F 3A 2A 12 :94
1A30 3D CD 7C 1A 81 32 28 12 3A 29 12 B7 C8 3D CD 7D :08
1A40 1A C6 03 4F 3A 2A 12 3D CD 7C 1A 81 32 29 12 C9 :FF
1A50 CD 20 1A 3A 2A 12 21 63 1A 3D 4F 06 00 09 7E 32 :66
1A60 2A 12 C9 0A 1A 7E CD 20 1A 3A 2C 12 D6 07 FE 08 :09
1A70 D2 B9 17 CD 7D 1A C6 46 32 2A 12 C9 07 07 07 07 :65
1A80 C9 53 54 41 52 54 2F 44 45 46 42 2F 44 45 46 57 :EC
1A90 2F 44 45 46 53 2F 44 45 46 4D 2F 4F 46 46 53 45 :3E
1AA0 54 2F 45 51 55 2F 49 46 2F 47 4F 54 4F 2F 4A 52 :5F
1AB0 2F 44 4A 4E 5A 2F 4A 4D 50 2F 43 41 4C 4C 2F 47 :3C
1AC0 4F 53 55 42 2F 44 4F 2F 55 4E 54 49 4C 2F 28 49 :56
1AD0 58 29 3D 28 49 58 29 2B 31 2F 28 49 59 29 3D 28 :93
1AE0 49 59 29 2B 31 2F 28 49 58 29 3D 28 49 58 29 2D :A4
1AF0 31 2F 28 49 59 29 3D 28 49 59 29 2D 31 2F 44 45 :99
------------------------------------------------------------
Sum: 38 09 97 6C 5C 0C ED 80 22 26 AE 27 BA 8F A7 A5 :CB


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1B00 3C 3E 48 4C 2F 48 4C 3C 3E 44 45 2F 41 46 3C 3E :04
1B10 41 46 27 2F 45 58 58 2F 28 53 50 29 3C 3E 48 4C :03
1B20 2F 44 45 43 49 4D 41 4C 2F 43 4D 50 4C 28 41 29 :0B
1B30 2F 43 4D 50 4C 28 43 59 29 2F 43 59 3D 31 2F 43 :F3
1B40 59 3D 30 2F 41 3D 30 2F 4E 4F 50 2F 48 41 4C 54 :17
1B50 2F 44 49 4E 54 2F 45 49 4E 54 2F 47 4F 54 4F 28 :4D
1B60 48 4C 29 2F 50 43 3D 48 4C 2F 53 50 3D 48 4C 2F :22
1B70 52 45 54 2F 52 45 54 55 52 4E 2F 42 3D 42 2B 31 :46
1B80 2F 43 3D 43 2B 31 2F 44 3D 44 2B 31 2F 45 3D 45 :94
1B90 2B 31 2F 48 3D 48 2B 31 2F 4C 3D 4C 2B 31 2F 28 :6B
1BA0 48 4C 29 3D 28 48 4C 29 2B 31 2F 42 3D 42 2D 31 :89
1BB0 2F 43 3D 43 2D 31 2F 44 3D 44 2D 31 2F 45 3D 45 :98
1BC0 2D 31 2F 48 3D 48 2D 31 2F 4C 3D 4C 2D 31 2F 28 :71
1BD0 48 4C 29 3D 28 48 4C 29 2D 31 2F 4E 45 47 2F 52 :C7
1BE0 45 54 49 2F 52 45 54 4E 2F 52 52 44 2F 52 4C 44 :72
1BF0 2F 4C 44 49 52 2F 4C 44 44 52 2F 4C 44 49 2F 4C :32
------------------------------------------------------------
Sum: B7 3D AE F1 06 FF 1C F3 9B 4F D7 23 C2 0C B5 BF :CD


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1C00 44 44 2F 43 50 49 52 2F 43 50 44 52 2F 43 50 49 :48
```

```
1C10 2F 43 50 44 2F 4F 55 54 49 52 2F 4F 55 54 44 52 :85
1C20 2F 4F 55 54 49 2F 4F 55 54 44 2F 49 4E 49 52 2F :6B
1C30 49 4E 44 52 2F 49 4E 49 2F 49 4E 44 2F 53 50 3D :55
1C40 49 58 2F 53 50 3D 49 59 2F 47 4F 54 4F 28 49 58 :83
1C50 29 2F 47 4F 54 4F 28 49 59 29 2F 50 43 3D 49 58 :24
1C60 2F 50 43 3D 49 59 2F 49 4D 4F 44 45 30 2F 49 4D :33
1C70 4F 44 45 31 2F 49 4D 4F 44 45 32 2F 28 53 50 29 :FB
1C80 3C 3E 49 58 2F 28 53 50 29 3C 3E 49 59 2F 41 3D :07
1C90 49 2F 49 3D 41 2F 58 48 3D 58 48 2B 31 2F 58 4C :1A
1CA0 3D 58 4C 2B 31 2F 59 48 3D 59 48 2B 31 2F 59 4C :1B
1CB0 3D 59 4C 2B 31 2F 58 48 3D 58 48 2D 31 2F 58 4C :1B
1CC0 3D 58 4C 2D 31 2F 59 48 3D 59 48 2D 31 2F 59 4C :1F
1CD0 3D 59 4C 2D 31 2F 48 4C 3D 44 45 2F 48 4C 3D 42 :0B
1CE0 43 2F 44 45 3D 48 4C 2F 42 43 3D 48 4C 2F 00 41 :C1
1CF0 3D 41 2B 2F 41 3D 41 2D 2F 41 3D 41 2E 26 2E 2F :63
-----------------------------------------------------------
Sum: D4 7E 47 F6 C5 D6 BB 73 F3 99 01 F7 CA A6 6F 4C :07
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1D00 41 3D 41 2E 41 4E 44 2E 2F 41 3D 41 2E 58 4F 52 :03
1D10 2E 2F 41 3D 41 2E 4F 52 2E 2F 43 50 28 41 2D 2F :A0
1D20 48 4C 3D 48 4C 2B 2E 2F 48 4C 3D 48 4C 2B 2F 48 :F4
1D30 4C 3D 48 4C 2D 2E 2F 48 4C 3D 48 4C 2D 2F 49 58 :09
1D40 3D 49 58 2B 2F 49 59 3D 49 59 2B 2F 52 4C 43 28 :1C
1D50 2F 52 52 43 28 2F 52 4C 28 2F 52 52 28 2F 53 4C :FC
1D60 41 28 2F 53 52 41 28 2F 53 4C 49 28 2F 53 52 4C :05
1D70 28 2F 42 49 54 28 2F 52 45 53 28 2F 53 45 54 28 :E2
1D80 2F 41 3D 50 4F 52 54 28 2F 50 4F 52 54 28 43 29 :22
1D90 3D 2F 50 4F 52 54 28 2F 21 28 2F 5B 2F 5D 2F 52 :E8
1DA0 45 53 54 41 52 54 28 2F 3B 2F 00 42 43 2F 44 45 :D1
1DB0 2F 48 4C 2F 53 50 2F 49 58 2F 49 59 2F 42 2F 43 :19
1DC0 2F 44 2F 45 2F 48 2F 4C 2F 28 48 4C 29 2F 41 2F :8C
1DD0 58 48 2F 58 4C 2F 28 49 58 29 2F 59 48 2F 59 4C :38
1DE0 2F 28 49 59 29 2F 00 3C 3E 2F 4E 5A 2F 3D 2F 5A :97
1DF0 2F 43 59 3D 30 2F 4E 43 2F 43 59 3D 31 2F 43 59 :FC
-----------------------------------------------------------
Sum: 9D E9 4F 4B 12 D5 6A E4 D1 B9 D8 81 91 C6 21 3A :EA
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1E00 2F 50 56 3D 30 2F 50 4F 2F 50 56 3D 31 2F 50 45 :17
1E10 2F 50 4C 55 53 2F 50 4C 55 53 2F 4D 49 4E 55 53 :A1
1E20 2F 4D 49 4E 55 53 2F 00 CE 1E E2 1E FD 1E 17 1F :27
1E30 25 1F 6A 1F 80 1F 8F 1F C6 22 DD 22 1A 23 D0 22 :30
1E40 1E 23 1E 23 2C 23 47 23 46 1F 4F 1F 58 1F 61 1F :05
1E50 EB EB 08 D9 E3 27 2F 3F 37 B7 AF 00 76 F3 FB E9 :19
1E60 E9 F9 C9 C9 04 0C 14 1C 24 2C 34 05 0D 15 1D 25 :A1
1E70 2D 35 ED 44 ED 4D ED 45 ED 67 ED 6F ED B0 ED B8 :F1
1E80 ED A0 ED A8 ED B1 ED B9 ED A1 ED A9 ED B3 ED BB :D2
1E90 ED A3 ED AB ED B2 ED BA ED A2 ED AA DD F9 FD F9 :60
1EA0 DD E9 FD E9 DD E9 FD E9 ED 46 ED 56 ED 5E DD E3 :D9
1EB0 FD E3 ED 57 ED 47 DD 24 DD 2C FD 24 FD 2C DD 25 :AE
1EC0 DD 2D FD 25 FD 2D 62 6B 60 69 54 5D 44 4D CD EE :E9
1ED0 17 CD 5D 17 2A 19 12 19 22 42 12 E5 DD E1 22 42 :43
1EE0 12 C9 CD EE 17 CD 5D 17 7B CD A1 23 2A 0F 12 7E :C3
1EF0 FE 3A 28 EE FE 20 C8 FE 0D C8 C3 B9 17 CD EE 17 :6C
-----------------------------------------------------------
Sum: 89 54 44 B3 38 39 22 96 54 41 F1 48 6F D5 85 3F :D3
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1F00 CD 5D 17 CD B4 23 2A 0F 12 7E FE 3A 28 EF FE 20 :1B
1F10 C8 FE 0D C8 C3 B9 17 CD EE 17 CD 5D 17 AF CD A1 :5E
1F20 23 1D 20 FA C9 CD EE 17 2A 0F 12 7E FE 22 C2 B9 :59
1F30 17 23 7E FE 0D CA B9 17 FE 22 28 05 CD A1 23 18 :53
1F40 F0 23 22 0F 12 C9 CD 0D 24 03 DD 34 00 00 C9 CD :C7
1F50 0D 24 03 FD 34 00 00 C9 CD 0D 24 03 DD 35 00 00 :41
1F60 C9 CD 0D 24 03 FD 35 00 00 C9 CD EE 17 CD 5D 17 :D8
1F70 ED 53 19 12 EB 7C 2F 67 7D 2F 6F 23 22 1B 12 C9 :BE
1F80 CD EE 17 CD 5D 17 2A 17 12 2B 2B 73 23 72 C9 CD :5A
1F90 EE 17 CD 98 1F C3 4D 21 CD 4C 19 B7 28 07 3D B7 :C6
1FA0 1F 32 21 12 C9 CD 31 16 41 3D 30 0D 00 20 0E E5 :2F
1FB0 21 21 12 36 01 E1 CD 0D 24 01 B7 00 C9 CD 31 16 :FF
1FC0 41 3C 3E 30 0D 00 20 0E E5 21 21 12 36 00 E1 CD :43
1FD0 0D 24 01 B7 00 C9 CD 52 19 B7 CA B9 17 F5 CD 31 :2E
1FE0 16 43 28 00 CA CC 20 F1 FE 0E 28 6C CD F1 1F 18 :BD
1FF0 41 FE 07 38 13 FE 0F D2 B9 17 D6 07 CD 7D 1A C6 :47
-----------------------------------------------------------
Sum: 22 FB 92 9B B1 D0 AA C5 8F 80 56 D7 1B 47 14 9A :86
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2000 04 CD A1 23 3C C3 A1 23 FE 04 CA B9 17 30 0B 3D :6C
2010 07 C6 78 5F C6 39 57 C3 B4 23 FE 06 20 0A CD 0D :9C
2020 24 04 FD 7C FD B5 00 C9 CD 0D 24 04 DD 7C DD B5 :09
2030 00 C9 CD 31 16 3C 3E 30 0D 00 20 08 E5 21 21 12 :F5
2040 36 00 E1 C9 CD 31 16 3D 30 0D 00 20 08 E5 21 21 :BD
2050 12 36 01 E1 C9 C3 B9 17 CD 31 16 3D 00 20 09 E5 :E5
2060 21 21 12 36 01 E1 18 35 CD 31 16 3C 3E 00 20 09 :70
2070 E5 21 21 12 36 00 E1 18 24 CD 31 16 3E 3D 00 20 :3B
2080 09 E5 21 21 12 36 02 E1 18 13 CD 31 16 3C 00 20 :F6
2090 09 E5 21 21 12 36 03 E1 18 03 C3 B9 17 CD 58 19 :48
20A0 B7 28 0E D6 07 DA B9 17 FE 08 30 11 C6 B8 C3 A1 :9D
20B0 23 CD 5D 17 3E FE CD A1 23 7B C3 A1 23 C6 07 CD :CD
20C0 BF 23 CD 03 24 BE 3A 25 12 C3 A1 23 F1 CD 52 19 :B5
20D0 F5 B7 CA B9 17 FE 07 30 25 3D FE 04 28 0E FE 05 :18
20E0 28 13 CD 7C 1A C6 0B CD A1 23 18 35 CD 0D 24 02 :4D
20F0 DD 2B 00 18 2C CD 0D 24 02 FD 2B 00 18 23 D6 07 :8C
-----------------------------------------------------------
Sum: 22 AF 09 A0 CC 55 E2 40 A5 29 CE 72 91 AB 8C 0E :A1
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2100 FE 08 30 0B CD 7D 1A C6 05 CD A1 23 F1 18 16 C6 :E6
2110 07 CD BF 23 CD 03 24 35 3A 25 12 CD A1 23 F1 18 :EA
2120 04 F1 CD F1 1F CD 31 16 29 3D 30 0D 00 20 08 E5 :96
2130 21 21 12 36 01 E1 C9 CD 31 16 29 3C 3E 30 0D 00 :29
2140 20 08 E5 21 21 12 36 00 E1 C9 C3 B9 17 CD EE 17 :A6
2150 CD 31 16 54 48 45 4E 0D 00 28 76 CD 31 16 47 4F :98
2160 54 4F 0D 00 CA 13 22 CD 31 16 47 4F 0D 00 20 0A :90
2170 E5 21 23 12 36 C2 E1 C3 FC 21 CD 31 16 47 4F 53 :F1
2180 55 42 0D 00 20 09 E5 21 23 12 36 C4 E1 18 6D CD :35
2190 31 16 43 41 4C 4C 0D 00 20 09 E5 21 23 12 36 C4 :CE
21A0 E1 18 59 CD 31 16 52 45 54 0D 00 20 0B 3A 21 12 :F6
21B0 CD 7D 1A C6 C0 C3 A1 23 CD 31 16 52 45 54 55 52 :17
21C0 4E 0D 00 C2 B9 17 3A 21 12 CD 7D 1A C6 C0 C3 A1 :A8
21D0 23 DD E5 E1 22 2E 12 3A 2D 12 B7 28 03 C3 B9 17 :16
21E0 E5 21 2D 12 36 01 E1 3A 21 12 FE 04 D2 B9 17 C3 :3B
21F0 7D 1A C6 20 EE 08 CD A1 23 DD 23 C9 3A 21 12 CD :07
-----------------------------------------------------------
Sum: 57 A2 94 85 7F D6 9E 3A 8E 94 DF A5 64 CA 7E CD :5E
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2200 7D 1A 4F 3A 23 12 81 CD A1 23 CD EE 17 CD 5D 17 :7A
2210 C3 B4 23 E5 21 23 12 36 C2 E1 E5 21 27 12 36 00 :23
2220 E1 3A 15 12 B7 28 D5 FE 01 28 25 FE 02 28 67 3A :0B
2230 27 12 B7 CA AF 22 2A 0F 12 7E FE 3A 20 05 36 20 :07
2240 C3 DD 22 CD 03 24 C3 CD EE 17 CD 5D 17 C3 B4 23 :26
2250 2A 0F 12 E5 CD EE 17 CD 5D 17 DD E5 E1 3A 21 12 :53
2260 FE 04 30 29 ED 4B 1B 12 09 23 EB B7 ED 52 7C B7 :00
2270 28 06 FE FF 28 10 18 15 7D FE 80 30 10 E1 36 3A :1C
2280 CD A1 23 C3 B4 23 7D FE 80 38 02 18 F0 E1 36 20 :9F
2290 CD A1 23 C3 B4 23 2A 0F 12 7E FE 3A 20 05 CD B4 :D2
22A0 23 18 06 CD A1 23 CD B4 23 CD EE 17 C3 5D 17 2A :A9
22B0 0F 12 7E FE 3A 20 0C 36 20 3A 21 12 CD 7D 1A C6 :F0
22C0 20 18 1C C3 FC 21 E5 21 27 12 36 01 E1 C3 21 22 :91
22D0 CD 03 24 C3 CD EE 17 CD 5D 17 C3 B4 23 3E 18 CD :87
22E0 A1 23 CD EE 17 CD 5D 17 3A 15 12 FE 03 C2 A1 23 :BF
22F0 DD E5 E1 ED 4B 1B 12 09 23 EB B7 ED 52 7C B7 28 :70
-----------------------------------------------------------
Sum: 92 9F 58 87 FD 6C 8A D6 FD DF BB 8B 4E 3B 7C 95 :95
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2300 07 FE FF 28 0C C3 BD 17 7D FE 80 D2 B9 17 C3 A1 :D0
2310 23 7D FE 80 DA BD 17 C3 A1 23 3E 10 18 C1 CD EE :35
2320 17 3E CD CD A1 23 CD 5D 17 C3 B4 23 CD EE 17 ED :4D
2330 5B 0F 12 1A FE 0D C4 E6 24 DD E5 D1 2A 40 12 2B :A9
2340 72 2B 73 22 40 12 C9 CD EE 17 CD 31 16 44 45 43 :FF
2350 28 42 29 3D 30 0D 00 28 21 CD 98 1F 3A 21 12 CD :14
2360 7D 1A C6 C2 EE 08 CD A1 23 2A 40 12 5E 23 56 23 :1C
2370 22 40 12 2A 1B 12 19 EB 18 3A 2A 40 12 5E 23 56 :74
2380 23 22 40 12 DD E5 E1 EB B7 ED 52 2B 2B 7C FE FF :EA
2390 C2 BD 17 7D FE 80 DA BD 17 CD 0D 24 01 10 0A 00 :58
23A0 C9 F5 3A 15 12 FE 03 20 07 F1 DD 77 00 DD 23 C9 :55
23B0 DD 23 F1 C9 F5 7B CD A1 23 7A CD A1 23 F1 C9 FE :7E
23C0 14 20 07 CD 03 24 FD 3E 0D C9 FE 11 20 07 CD 03 :46
23D0 24 DD 3E 0D C9 C3 B9 17 FE 05 20 07 CD 03 24 DD :A3
23E0 3E 03 C9 FE 06 20 07 CD 03 24 FD 3E 03 C9 C3 B9 :AC
23F0 17 FE 12 38 07 CD 03 24 FD D6 07 C9 CD 03 24 DD :CE
-----------------------------------------------------------
Sum: ED 84 F2 57 B9 9B 5F 4D A6 F6 51 FE 94 1C 55 6C :16
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2400 D6 04 C9 E3 F5 7E 23 CD A1 23 F1 E3 C9 E3 F5 7E :A0
2410 23 B7 20 03 F1 E3 C9 FE 0A 38 2B 20 04 F1 F5 18 :27
2420 31 FE 0B 20 03 78 18 2A FE 0C 20 03 79 18 23 FE :F6
2430 0D 20 05 CD B4 23 18 D7 CD 26 16 47 45 4E 20 45 :0D
2440 52 52 00 C3 CB 17 F5 7E 23 CD A1 23 F1 3D 20 F6 :B4
2450 18 BD CD A1 23 18 B8 2A 0F 12 7E FE 0D 28 24 FE :54
2460 3B 28 20 FE 20 28 4D CD CF 24 2A 0F 12 7E 23 FE :C0
2470 3A 28 0A FE 20 28 06 FE 0D 28 08 18 F0 2B 22 0F :57
2480 12 18 31 CD 0A 18 2A 0F 12 22 07 12 3A 2D 12 B7 :00
2490 C8 E5 21 2D 12 36 00 E1 DD E5 E1 ED 5B 2E 12 13 :62
24A0 13 B7 ED 52 7D EB 2B DD E5 E5 DD E1 CD A1 23 DD :6F
24B0 E1 C9 77 C9 CD EE 17 2A 0F 12 7E FE 3B 28 C4 CD :77
24C0 E6 24 CD EE 17 2A 0F 12 7E FE 0D 28 B6 18 E5 ED :78
24D0 5B 17 12 DD E5 E1 ED 4B 1B 12 09 7D 12 13 7C 12 :C5
24E0 13 ED 53 17 12 C9 CD 3E 19 B7 28 37 FE 15 38 14 :DE
24F0 FE 37 30 1E 21 50 1E D6 15 4F 06 00 09 7E CD A1 :47
-----------------------------------------------------------
Sum: 36 14 08 48 60 C6 6F A7 2E CC 2A 4F F7 2A 27 02 :93
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2500 23 C3 EE 17 3D 6F 26 00 29 11 28 1E 19 5E 23 56 :2D
2510 EB E9 D6 37 6F 26 00 29 01 72 1E 09 5E 23 56 CD :DD
2520 B4 23 C9 CD 89 19 B7 CA 5D 28 FE 01 C2 C3 25 E5 :A3
2530 21 27 12 36 80 E1 E5 21 28 12 36 00 E1 CD 31 16 :5C
2540 2E 00 20 07 E5 21 28 12 36 08 E1 CD 31 16 31 0D :06
2550 00 28 5A CD 58 19 B7 28 1D FE 07 DA B9 17 FE 0F :78
2560 30 28 D6 07 4F 3A 27 12 81 4F 3A 28 12 81 C3 A1 :20
2570 23 11 01 00 18 03 CD 5D 17 3A 27 12 4F 3A 28 12 :C7
2580 81 C6 46 CD A1 23 7B C3 A1 23 FE 12 38 08 3E FD :AB
2590 CD A1 23 79 18 08 3E DD CD A1 23 79 C6 03 D6 07 :F5
25A0 FE 0D 20 09 CD 62 25 3A 25 12 C3 A1 23 3A 28 12 :F4
25B0 FE 08 28 BD 3A 27 12 E6 10 0F 0F 0F 0F 0E 3C 81 :5B
25C0 C3 A1 23 21 27 12 36 90 23 36 00 2B FE 02 CA 3D :32
25D0 25 36 A0 FE 05 CA 53 25 36 A8 CA 53 25 36 B0 FE :54
25E0 06 CA 53 25 36 B8 FE 07 20 10 CD 52 19 CD 56 25 :EB
25F0 CD 31 16 29 0D 00 C8 C3 B9 17 23 23 FE 0E D2 9D :66
-----------------------------------------------------------
Sum: 69 A5 CD A5 88 5E D4 FC 6F 36 70 37 CF 5F 03 81 :34
```

BASE FM-8 ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2600 26 11 00 00 ED 53 27 12 36 00 D6 08 28 22 3D 28 :73
2610 1B 3D 28 35 3D 28 22 3D 28 09 36 09 3E FD 32 27 :7D
2620 12 18 28 36 09 3E DD 32 27 12 18 1F 36 09 18 1B :C0
2630 36 4A 3E ED 32 28 12 18 12 3E B7 CD A1 23 36 42 :3F
2640 E5 21 28 12 36 ED E1 18 02 18 F3 CD 58 19 B7 28 :86
2650 26 FE 07 D2 B9 17 3A 27 12 B7 C4 A1 23 3A 28 12 :F3
2660 B7 C4 A1 23 79 FE 05 38 02 3E 03 3D CD 7C 1A 4F :25
2670 3A 29 12 81 C3 A1 23 CD 5D 17 CD 0D 24 02 D5 11 :A4
2680 0D 00 3A 27 12 B7 C4 A1 23 3A 28 12 B7 C4 A1 23 :72
2690 3A 29 12 C6 10 CD A1 23 3E D1 C3 A1 23 FE 16 30 :B6
26A0 5B D6 0E 32 27 12 CD 52 19 B7 CA B9 17 FE 07 DA :12
26B0 B9 17 FE 0F 30 32 D6 07 32 28 12 FE 07 28 1C CD :9E
26C0 03 24 CB 3A 28 12 4F 3A 27 12 CD 7D 1A 81 CD A1 :7B
26D0 23 CD 31 16 29 0D 00 C2 B9 17 C9 3A 27 12 FE 04 :3D
26E0 38 E1 CD 03 24 CB 18 DB CD BF 23 D6 07 32 28 12 :C3
26F0 CD 03 24 CB 3A 25 12 CD A1 23 18 C7 FE 19 30 65 :4C
---------------------------------------------------------
Sum: 0B A7 B5 2C BB 5B FC 9E 04 72 FA 73 E7 E2 88 5C :D0
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2700 D6 16 0F 0F C6 40 32 27 12 2A 0F 12 7E 23 22 0F :98
2710 12 FE 30 DA B9 17 FE 38 D2 B9 17 D6 30 CD 7D 1A :2C
2720 32 28 12 CD 31 16 2C 00 C2 B9 17 CD 52 19 FE 07 :7B
2730 DA B9 17 FE 0F 30 1D CD 03 24 CB D6 07 4F 3A 27 :50
2740 12 81 4F 3A 28 12 81 CD A1 23 CD 31 16 29 0D 00 :B2
2750 C8 C3 B9 17 CD BF 23 F5 CD 03 24 CB 3A 25 12 CD :FC
2760 A1 23 F1 18 D6 FE 1C 30 62 FE 19 28 1C FE 1A 28 :EA
2770 3E CD 5D 17 CD 03 24 D3 D5 CD 31 16 29 3D 41 0D :E3
2780 00 C2 B9 17 D1 7B C3 A1 23 CD 31 16 43 29 0D 00 :F2
2790 28 15 CD 5D 17 7B CD 0D 24 01 DB 0A 00 CD 31 16 :F1
27A0 29 0D 00 C8 C3 B9 17 CD 0D 24 02 ED 78 00 C9 CD :8C
27B0 58 19 FE 07 DA B9 17 FE 0F D2 B9 17 D6 07 CD 7D :F6
27C0 1A C6 41 CD 0D 24 01 ED 0A 00 C9 FE 1C 20 29 2A :6D
27D0 0F 12 7E FE 22 CC 28 1F 7E FE 20 CC EE 17 CD 31 :3D
27E0 16 29 0D 00 C8 2A 0F 12 EB CD BC 0B DA B9 17 EB :73
27F0 CD A1 23 22 0F 12 18 D7 FE 1F 30 3E D6 1D 28 07 :70
---------------------------------------------------------
Sum: 62 C8 31 64 E2 03 6B 5F 22 5F DF FC E7 EB 5A 06 :FC
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2800 3E C1 32 27 12 18 05 3E C5 32 27 12 CD 31 16 41 :4A
2810 46 0D 00 28 12 CD 58 19 CA B9 17 FE 04 CA B9 17 :01
2820 FE 07 D2 B9 17 18 02 3E 04 FE 05 D4 D8 23 3D CD :DF
2830 7C 1A 4F 3A 27 12 81 C3 A1 23 FE 20 28 1B CD 5D :EB
2840 17 7B FE 08 D2 B9 17 CD 7D 1A C6 C7 CD A1 23 CD :89
2850 31 16 29 0D 00 C8 C3 B9 17 CD 0A 18 C9 CD 31 16 :A4
2860 21 00 20 1B CD 03 24 CD CD 5D 17 C3 B4 23 06 00 :FE
2870 FE 07 DA D8 23 FE 0F 38 05 CD BF 23 06 01 C9 CD :70
2880 52 19 B7 CA C7 2A 32 27 12 32 2A 12 CD 31 16 2B :F5
2890 0D 00 3A 27 12 20 2A 0E 03 2E 04 FE 05 D4 6E 28 :7A
28A0 3D FE 05 30 07 CD 7C 1A 81 C3 A1 23 05 20 09 CD :DD
28B0 B8 28 3A 25 12 C3 A1 23 D6 06 CD 7D 1A 85 C3 A1 :01
28C0 23 CD 31 16 2D 0D 00 3A 27 12 20 06 0E 0B 2E 05 :56
28D0 18 C9 CD 31 16 2B 2B 0D 00 3A 27 12 20 0C 0E 03 :08
28E0 2E 04 CD 9B 28 3A 27 12 18 B1 CD 31 16 2D 2D 0D :79
28F0 00 3A 27 12 20 06 0E 0B 2E 05 18 E6 E5 21 28 12 :23
---------------------------------------------------------
Sum: 22 9A 96 84 A1 E3 C6 B9 73 48 AF A8 3B DA DD 1A :F7
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2900 36 00 E1 CD 31 16 3D 00 C2 B9 17 3A 27 12 FE 0F :7A
2910 38 30 3E 01 32 28 12 3A 25 12 B7 28 0B F5 CD 58 :88
2920 19 F1 32 25 12 79 18 03 CD 58 19 B7 28 76 32 29 :F5
2930 12 32 28 12 3A 27 12 CD BF 23 32 27 12 3A 29 12 :80
2940 18 09 CD 58 19 B7 28 5C 32 29 12 FE 0F 38 13 3A :99
2950 28 12 B7 C2 B9 17 3A 29 12 CD BF 23 32 29 12 32 :46
2960 28 12 3A 27 12 D6 07 DA B9 17 CD 7D 1A 4F 3A 29 :4A
2970 12 D6 07 DA B9 17 81 C6 40 CD A1 23 3A 28 12 B7 :DC
2980 C8 3A 27 12 D6 0D 28 0E 3A 29 12 D6 0D 20 06 3A :0C
2990 25 12 C3 A1 23 C9 3A 29 12 D6 0D CA B9 17 3A 25 :D8
29A0 12 C3 A1 23 CD 31 16 50 4F 52 54 28 43 29 0D 00 :93
29B0 20 15 3A 27 12 D6 07 DA B9 17 CD 7D 1A C6 40 CD :66
29C0 0D 24 01 ED 0A 00 C9 CD 31 16 28 00 28 50 CD 5D :D0
29D0 17 3A 2A 12 FE 07 30 10 FE 05 D4 D8 23 3D CD 7C :2A
29E0 1A 3C CD A1 23 C3 B4 23 FE 0F 30 0E D6 07 CD 7D :F3
29F0 1A C6 06 CD A1 23 7B C3 A1 23 FE 11 28 09 FE 14 :CB
---------------------------------------------------------
Sum: 8A DA 01 8A F0 63 0A 53 D2 D5 C2 3D 6D 52 89 84 :11
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2A00 28 05 CD BF 23 18 E5 CD BF 23 D6 07 CD 7D 1A C6 :8F
2A10 06 CD A1 23 3A 25 12 CD A1 23 7B C3 A1 23 3A 2A :FF
2A20 12 32 2C 12 AF 32 1F 12 CD 8F 19 28 41 CD 5D 17 :B3
2A30 D5 CD 31 16 29 0D 00 C2 B9 17 D1 3A 2C 12 FE 0E :06
2A40 20 08 3E 3A CD A1 23 C3 B4 23 FE 04 28 11 FE 03 :07
2A50 38 0D FE 05 D4 D8 23 CD 0D 24 01 2A 0D 00 C9 3D :53
2A60 CD 7C 1A C6 4B CD 0D 24 01 ED 0A 0D 00 C9 3A 2C :A6
2A70 12 FE 0F D2 B9 17 D6 07 DA B9 17 FE 07 20 43 CD :7D
2A80 50 1A 3A 28 12 B7 28 0D 3A 2B 12 B7 C4 A1 23 3A :BA
2A90 28 12 CD A1 23 3A 2B 12 B7 28 0F CD A1 23 3A 2A :25
2AA0 12 CD A1 23 AF CD A1 23 18 06 3A 2A 12 CD A1 23 :08
2AB0 3A 29 12 B7 C8 3A 2B 12 B7 C4 A1 23 3A 29 12 C3 :E2
2AC0 A1 23 CD 66 1A 18 BB CD 31 16 28 00 C2 B9 17 3E :F0
2AD0 01 32 1F 12 CD 8F 19 28 4C CD 5D 17 D5 CD 31 16 :77
2AE0 29 3D 00 C2 B9 17 CD 58 19 B7 CA 69 2B FE 0E 20 :77
2AF0 09 D1 CD 0D 24 01 32 0D 00 C9 FE 07 D2 B9 17 FE :86
---------------------------------------------------------
Sum: E4 E5 A3 CB 4A 90 31 D7 D8 59 A4 BD 5C 70 70 0A :F1
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2B00 03 28 14 D1 FE 05 30 18 3D CD 7C 1A C6 43 CD 0D :DE
2B10 24 01 ED 0A 0D 00 C9 D1 CD 0D 24 01 22 0D 00 C9 :BA
2B20 CD D8 23 18 F3 CD 58 19 B7 CA B9 17 32 2C 12 FE :D0
2B30 0E 28 22 FE 07 DA B9 17 FE 0F D2 B9 17 3A 2A 12 :2C
2B40 FE 02 DA B9 17 CD 66 1A 3A 2C 12 D6 07 C6 70 32 :B4
2B50 2A 12 C3 82 2A CD 50 1A 3A 2A 12 D6 08 FE 76 20 :CA
2B60 02 3E 77 32 2A 12 C3 82 2A D5 CD 5D 17 43 D1 D1 :8F
2B70 CD 0D 24 02 E5 21 0D 01 36 0B 01 E1 00 C9 2A 09 :33
2B80 12 22 07 12 CD 26 16 4C 41 42 45 4C 20 54 41 42 :AD
2B90 4C 45 20 46 52 4F 4D 20 3F 0D 00 11 A3 11 CD 10 :F3
2BA0 31 CD B9 0B DA B9 2D 22 11 12 CD 26 16 2A 50 41 :8B
2BB0 53 53 20 30 0D 00 EB 2A 07 12 7E FE 3B 28 5F FE :6D
2BC0 0D 28 5B FE 20 28 57 FE 3A 28 53 B7 28 5E ED 53 :5D
2BD0 1F 12 FE 2B CA C5 17 FE 2F CA C5 17 FE 2D CA C5 :8D
2BE0 17 FE 3A 28 0E FE 20 28 0A FE 0D 28 06 12 13 23 :56
2BF0 7E 18 DF 3E 0D 12 13 D5 AF 12 2A 11 12 ED 5B 1F :2F
---------------------------------------------------------
Sum: 9C 5F F0 82 60 A4 AC 81 4D 5E FC 5D A9 C7 CC FD :DB
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2C00 12 1A 13 46 23 04 05 28 10 B7 CA C9 17 B8 28 F1 :1B
2C10 2B 7E 23 FE 0D 28 E6 18 F8 B7 C2 C9 17 D1 2A 07 :50
2C20 12 7E 23 FE 0D 20 FA 22 07 12 18 8B 12 13 ED 53 :1B
2C30 13 12 ED 53 17 12 C9 2A 09 12 22 07 12 22 0F 12 :1A
2C40 E5 21 2D 12 36 00 E1 2A 13 12 22 17 12 21 40 12 :69
2C50 22 40 12 2A 07 12 7E B7 C8 DD 22 42 12 2A 07 12 :4A
2C60 22 1D 12 CD 57 24 CD 95 0B FE 58 CA B9 2D FE 20 :2A
2C70 CC 70 12 DD E5 D1 2A 42 12 EB B7 ED 52 44 4D CD :9E
2C80 84 2C 18 CF 3A 15 12 FE 03 C0 3A 44 12 FE 01 C0 :08
2C90 CD A0 2C F5 CD C7 2C F1 C8 CD A0 2C C2 99 2C C9 :F0
2CA0 CD 47 12 2A 42 12 E5 ED 5B 1B 12 19 CD 91 12 E1 :68
2CB0 0C 0D C8 06 06 CD 6C 12 7E 23 22 42 12 CD 96 12 :C4
2CC0 0D C8 10 F1 F6 FF C9 3E 19 CD F0 2C 2A 0D 12 23 :40
2CD0 22 0D 12 EB CD E7 12 CD 6C 12 ED 5B 1D 12 1A FE :CC
2CE0 20 CC E8 2C CD 76 12 C9 06 05 CD 6C 12 10 FB C9 :48
2CF0 C5 47 CD 6C 12 3A 71 11 B8 C2 F2 2C C1 C9 E5 21 :3B
---------------------------------------------------------
Sum: 95 1E 9E E3 BE B6 F1 17 F7 DB C3 1E 4E 67 C1 F5 :CE
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2D00 15 12 36 00 E1 21 00 00 22 19 12 22 1B 12 22 0D :2A
2D10 12 21 40 12 22 40 12 CD 7E 2B CD 26 16 2A 50 41 :33
2D20 53 53 20 00 3A 15 12 C6 31 CD 59 12 CD 4C 12 CD :4E
2D30 37 2C 3A 15 12 3C 32 15 12 FE 04 C2 1A 2D DD E5 :26
2D40 E1 CD 47 12 11 51 2D CD 76 12 CD 91 12 CD 47 12 :81
2D50 C9 2A 4F 42 4A 45 43 54 20 45 4E 44 3A 0D CD 26 :DB
2D60 16 0C 20 20 23 23 23 23 20 42 41 53 45 20 46 4D :DC
2D70 2D 38 20 56 31 20 23 23 23 23 0D 0D 43 4F 50 59 :0D
2D80 52 49 47 48 54 28 43 29 20 42 59 20 43 41 52 52 :15
2D90 59 20 4C 41 42 2E 0D 00 AF 32 45 12 21 00 34 22 :32
2DA0 09 12 CD 47 14 3A 06 12 B7 20 0E E5 21 0F E0 36 :A5
2DB0 8A E1 E5 21 0E E0 36 07 E1 CD 4C 12 31 F0 10 11 :EA
2DC0 A3 11 CD 10 31 1A FE 45 20 05 CD 81 15 18 EA FE :A7
2DD0 4A 28 72 FE 4D CA 51 2E FE 23 CA 12 30 FE 4B CA :B8
2DE0 EB 2E FE 4F CA 7F 2E FE 21 CA 00 00 FE 41 28 42 :6F
2DF0 FE 24 28 02 18 C3 CD FB 2D 18 BE 1A 13 FE 24 C0 :01
---------------------------------------------------------
Sum: B2 D4 50 41 16 21 E2 BD 8F 36 F2 27 F8 93 02 63 :BB
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2E00 CD 04 31 1B D8 1A FE 0D 20 13 3E 24 CD 59 12 CD :B4
2E10 91 12 CD 6C 12 11 A3 11 CD 83 0B 18 DE FE 20 20 :42
2E20 01 13 CD BC 0B D8 77 23 18 DB 11 A3 11 CD 83 0B :2D
2E30 18 C4 13 1A FE 2F 3E 00 20 02 3E 01 32 44 12 CD :2A
2E40 FE 2C C3 B9 2D 13 CD B9 0B DA B9 2D 11 B9 2D D5 :03
2E50 E9 13 CD B9 0B CD 4C 12 3E 24 CD 59 12 CD 91 12 :C2
2E60 06 08 CD 6C 12 7E 23 CD 96 12 10 F6 CD 95 0B B7 :99
2E70 C4 70 12 FE 58 CA B9 2D FE 03 CA B9 2D 18 D6 13 :FE
2E80 1A FE 0D 28 0E ED 53 0F 12 CD 5D 17 EB CD 91 12 :58
2E90 C3 B9 2D ED 5B 11 12 DD 2A 13 12 1A B7 CA B9 2D :C1
2EA0 CD DE 2E DD 7E 00 DD 23 6F DD 7E 00 DD 23 67 CD :32
2EB0 91 12 3E 3A CD 80 12 D5 CD 76 12 D1 CD D7 2E CD :14
2EC0 95 0B B7 C4 70 12 FE 1B CA B9 2D FE 58 CA B9 2D :6C
2ED0 1A B7 CA B9 2D 18 C4 1A 13 FE 0D 20 FA C9 3A 71 :23
2EE0 11 B7 C8 FE 14 C8 CD 6C 12 18 F3 13 CD B9 0B CD :31
2EF0 91 12 CD 6C 12 7E CD 96 12 CD 6C 12 11 A3 11 CD :BE
---------------------------------------------------------
Sum: B4 D6 09 4C 0C 48 FR 21 7B 55 90 5A 87 1B 54 87 :86
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2F00 10 31 CD B9 0B 11 AB 11 1A FE 58 CA B9 2D FE 0D :CA
2F10 20 03 23 18 DA FE 2F 20 03 2B 18 D3 CD BC 0B 38 :6A
2F20 CE 77 23 18 F7 32 46 12 D5 CD 13 14 D1 21 F0 10 :BC
2F30 06 10 36 02 23 1A 13 77 23 10 FA 36 0D 2A 09 12 :CA
2F40 22 04 11 ED 5B 0B 12 EB B7 ED 52 23 22 02 11 21 :F6
2F50 00 00 22 06 11 3A 46 12 FE 01 28 16 CD 9B 0B 38 :B3
2F60 16 CD 9E 0B 38 11 CD AA 0B CD 26 16 0D 4F 4B 0D :14
2F70 00 C9 CD 08 C0 18 F2 CD 26 16 0D 45 52 52 0D 00 :74
2F80 C9 11 A5 11 1A FE 0D 20 58 CD A1 0B 38 30 11 38 :57
2F90 01 CD FF 2F 2A 09 12 22 04 11 ED 5B 02 11 19 23 :0F
2FA0 7C FE C0 30 25 3E 55 77 7E FE 55 20 1D CD A4 0B :23
2FB0 38 0C CD AA 0B CD 26 16 0D 4F 4B 0D 00 C9 CD 26 :3F
2FC0 16 0D 42 52 45 41 4B 0D 00 C9 CD 26 16 0D 4D 45 :06
2FD0 4D 20 4F 56 45 52 0D 00 C9 11 31 01 CD FF 2F 18 :D5
2FE0 A8 D5 CD A1 0B 30 03 D1 18 D4 11 31 01 CD FF 2F :24
2FF0 D1 D5 21 F1 10 06 10 CD 00 33 20 E6 D1 18 8F CD :29
---------------------------------------------------------
Sum: 96 14 97 45 7C A4 4F A8 C3 E3 87 4C BE 3A 1B B2 :DB
```

BASE FM-8 ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3000 4C 12 E5 21 01 11 36 0D E1 CD 92 0B 11 F1 10 C3 :D9
3010 92 0B CD 18 30 C3 B9 2D 3A 45 12 B7 20 15 3E 01 :17
3020 32 45 12 CD 26 16 50 52 49 4E 54 45 52 20 4F 4E :73
3030 0D 00 C9 AF 32 45 12 CD 26 16 50 52 49 4E 54 45 :E9
3040 52 20 4F 46 46 0D 00 C9 FE 0D 20 04 3E 0A 18 05 :B7
3050 FE 20 30 01 AF C3 D4 0B 0E E0 36 03 E1 E5 21 0E :BC
3060 E0 36 07 E1 C9 3A 0D E0 E6 07 FE 06 C8 18 F6 00 :B5
3070 AF CD 83 30 F1 D3 FF 3E 80 D3 FE 07 CD 83 30 AF :B7
3080 D3 FE C9 C5 D5 57 1E 06 01 00 00 DB FE E6 0D BA :36
3090 20 03 D1 C1 C9 0B 78 B1 20 F1 1D 20 EE CD 26 16 :F7
30A0 0D 2A 50 52 49 4E 54 45 52 20 45 52 52 0D 00 E5 :56
30B0 21 45 12 36 00 E1 C3 B9 2D 13 CD 04 31 22 04 11 :84
30C0 CD 04 31 ED 4B 04 11 B7 ED 42 23 22 02 11 CD 04 :5E
30D0 31 22 06 11 CD 04 31 22 1F 12 01 10 00 21 F1 10 :F2
30E0 EB ED B0 E5 21 01 11 36 0D E1 CD 9B 0B DA B9 2D :F7
30F0 2A 1F 12 22 04 11 CD 9E 0B DA B9 2D CD 26 16 0D :DE
---------------------------------------------------------
Sum: 30 47 8B 20 5C B7 FE AD C0 70 73 B8 C9 12 14 2D :57
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3100 4F 4B 0D 00 CD B9 0B 13 13 13 13 13 DA B9 2D C9 :20
3110 C3 83 0B D5 AF 32 93 11 CD 85 31 47 3A 9D 11 B7 :14
3120 CC 2F 32 78 E6 F0 FE C0 20 50 78 FE CD CA 5B 08 :19
3130 FE C9 28 1B FE CA 28 17 FE CB CA B3 08 FE C8 28 :4D
3140 19 FE C7 28 0A 3A 93 11 B7 FA 63 31 20 06 78 CD :9E
3150 BF 0B 18 C4 3D 32 93 11 18 09 3A 93 11 3C 32 93 :B9
3160 11 18 EB 78 CD D7 0B CD 8F 0B FE 62 20 AA 3A 93 :99
3170 11 E6 80 EE 80 32 93 11 18 9E 3A 93 11 B7 FA 63 :63
3180 31 28 E0 18 CF C3 C2 0B CD B1 0F CD D7 0B 7E 32 :9C
3190 8E 11 22 8F 11 21 92 11 36 43 3E 00 32 00 E0 32 :20
31A0 91 11 2F 32 00 E0 16 14 CD FF 09 CD 50 0A 78 07 :88
31B0 30 3D F5 CD CA 08 47 3A 4B 32 B8 28 1B F1 CD 32 :EA
31C0 0A FE 01 20 04 3E 01 18 02 3E 00 32 4D 32 47 3A :F6
31D0 4C 32 A8 B7 28 1C 18 CE 3A 4A 32 FE F0 20 03 F1 :BF
31E0 18 17 C5 06 00 10 FE C1 3C 32 4A 32 F1 18 B7 15 :88
31F0 20 B6 E5 21 4A 32 36 00 E1 CD FF 09 CD CA 08 FE :E1
---------------------------------------------------------
Sum: E4 51 35 5E 14 82 86 0C E8 0B E4 F1 BA FB EB E1 :39
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3200 F0 20 09 E5 21 4A 32 36 00 E1 18 ED F5 CD D7 0B :5B
3210 3A 8E 11 2A 8F 11 77 CD 32 0A FE 08 20 04 3E 00 :8B
3220 18 02 3E 01 32 4C 32 F1 32 4B 32 E1 D1 C1 C9 E5 :CA
3230 D5 C5 21 00 03 22 A1 11 CD D7 0B 01 E8 03 0B 78 :B0
3240 B1 20 FB CD D7 0B C1 D1 E1 C9 00 CD 01 01 3D 32 :F5
3250 30 20 47 4F 54 4F 20 43 50 52 45 47 30 33 0D 20 :AA
3260 43 3D 30 20 41 3D 30 3A 52 45 54 0D 43 50 52 45 :DA
3270 47 30 33 20 5B 42 43 20 21 44 45 5B 41 44 52 53 :F9
3280 5D 3A 41 3D 45 3A 28 4F 46 53 29 3D 41 0D 20 21 :99
3290 43 50 53 54 52 20 21 28 22 29 22 30 30 29 0D 20 :18
32A0 49 46 20 3C 3E 20 54 48 45 4E 20 5D 42 43 3A 43 :F7
32B0 3D 30 3A 41 3D 43 3A 52 45 54 0D 20 5D 42 43 3A :D6
32C0 41 3D 43 3A 52 45 54 0D 43 50 52 45 47 30 32 20 :E6
32D0 5D 44 45 20 28 43 50 27 29 3D 44 45 20 21 43 50 :AB
32E0 52 45 47 30 30 3A 52 45 54 0D 0D 43 50 52 45 47 :EE
32F0 32 20 48 4C 3D 52 54 41 42 4C 45 20 4A 52 20 43 :FC
---------------------------------------------------------
Sum: CA 08 23 50 A5 73 F1 3E C9 B5 91 2A 94 0D 5B 0A :CB
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3300 C5 D5 E5 1A BE 20 0B 05 28 08 FE 0D 28 04 13 23 :24
3310 18 F1 E1 D1 C1 C9 00 00 00 00 00 00 00 00 00 00 :45
3320 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
3330 42 41 53 45 20 46 4D 2D 38 20 56 45 52 20 31 20 :B1
3340 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
3350 20 20 20 20 31 32 30 30 2D 33 33 46 46 20 20 20 :C2
3360 48 4F 54 20 53 54 41 52 54 20 31 32 30 30 20 20 :BC
3370 43 4F 4C 44 20 53 54 41 52 54 20 31 32 30 33 20 :D6
3380 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
3390 31 39 38 32 20 4D 41 59 20 36 54 48 20 20 20 20 :4D
33A0 4F 52 49 47 49 4E 41 4C 20 28 43 29 20 20 20 20 :89
33B0 20 20 20 20 20 20 43 41 52 52 59 20 4C 41 42 2E :5E
33C0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 :00
33D0 46 4D 2D 38 20 56 65 72 73 69 6F 6E 20 20 20 20 :7E
33E0 20 45 54 46 20 4C 41 42 4F 52 41 54 4F 52 59 2E :4C
33F0 20 20 20 54 2E 4B 41 53 41 53 41 4B 55 20 20 20 :96
---------------------------------------------------------
Sum: 10 42 3B 3F 5A D0 E9 02 E8 AD D9 B9 92 D7 F2 9F :02
```

Z80 システム・モニタ アセンブル・リスト

```
0000                 ;*********************************
0000                 ;* Z-80 System Monitor           *
0000                 ;*  ♥♥♥♥ FM-8 Version 1.0 ♥♥♥♥   *
0000                 ;* Copyright (C) 1982            *
0000                 ;*      By HIRO/ETF Laboratory   *
0000                 ;* Written by T.Kasasaku.        *
0000                 ;*********************************
0000                 ;
0000   P             RCB:    EQU   0BE0H        ;REQUEST CODE BLOCK
0000   P             RCB8:   EQU   E808H
0000   P             BUFF:   EQU   043DH        ;CMT & KEYIN BUFFR
0000   P             BUF:    EQU   3D04H
0000   P             ATCODE: EQU   08FCH
0000   P             RETFLG: EQU   08FDH        ;RETURN MODE
0000                 ;
0000                         REL   0B80H
0B80                 ;
0B80                 ; Jump Table
0B80                 ;
0B80 C32E0E          MAIN:   JP    MAIN1        ;JUMP TO MONITOR
0B83 C3940D          GETL:   JP    GETL1        ;LINE INPUT
0B86 C3D10C          LETNL:  JP    LETNL1       ;RETURN & LINE FEED
0B89 C3CC0C          NL:     JP    NL1          ;CRLF IF CUR X<>0
0B8C C3980C          PRNTS:  JP    PRNTS1       ;SPACE PRINT
0B8F C39E0C          PRNT:   JP    PRNT1        ;ACC PRINT
0B92 C3BE0C          MSG:    JP    MSG1
0B95 C3470D          GETKY:  JP    GETKY1       ;REAL TIME KEYIN
0B98 C3FA0C          BRKEY:  JP    BRKEY1       ;BREAK CHECK
0B9B C3020C          ?WR1:   JP    B1SAVE
0B9E C30C0C          ?WRD:   JP    SAVE
0BA1 C3040C          ?RD1:   JP    B1LOAD
0BA4 C3080C          ?RDD:   JP    B2LOAD
0BA7 C3100C          ?VRFY:  JP    VERIFY
0BAA C39C0C          BELL:   JP    BELL1
0BAD C3C70D          HEX4PR: JP    H4PR1
0BB0 C3CC0D          HEX2PR: JP    H2PR1
0BB3 C3E20D          ASC:    JP    ASC1
0BB6 C3EB0D          HEX:    JP    HEX1
0BB9 C3000E          HLHEX:  JP    HLHEX1
0BBC C30F0E          2HEX:   JP    2HEX1
0BBF C3230C          ?DPCT:  JP    ?DPCT1
0BC2 C3000D          ??KEY:  JP    ??KEY1
0BC5 C3260E          RETURN: JP    RET1         ;RETURN TO F-BASIC
0BC8 C3A50D          GETC:   JP    GETC1
```

Z80 システム・モニタ アセンブル・リスト

```
OBCB C3F40C     BIOS:    JP      BIOS1           ;BIOS CALL (6809)
OBCE C3E80B     SCREEN:  JP      SCRN
OBD1 C3800C     WIDTH:   JP      WIDTH1
OBD4 C3280D     LPRNT:   JP      LPRNT1
OBD7 C3020C     AUX1:    JP      B1SAVE
OBDA C3020C     AUX2:    JP      B1SAVE
OBDD                     ;***********************************
OBDD                     ;* ♥ IOCS & Subroutine Package ♥ *
OBDD                     ;*  Copyright (C) 1982 ETF Lab.   *
OBDD                     ;***********************************
OBDD                     ;
OBDD            CRDATA:  ENT
OBDD 00                  DEFB    0
OBDE 00                  DEFB    0
OBDF 01                  DEFB    1
OBE0 00                  DEFB    0               ;BACK COLOR
OBE1 28         WDATA:   DEFB    40              ;WINDOW X
OBE2 14                  DEFB    20              ;WINDOW Y
OBE3 00                  DEFB    0               ;SCROLL FROM
OBE4 14                  DEFB    20              ;SCROLL LINES
OBE5 00                  DEFB    0               ;FUNCTION SWITCH
OBE6 01                  DEFB    1
OBE7 00                  DEFB    0               ;COLOR SWITCH
OBE8                     ;
OBE8            SCRN:    ENT
OBE8 22EB08              LD      (RCB+11),HL
OBEB 22ED08              LD      (RCB+13),HL
OBEE 3E11                LD      A,11H
OBF0 210707              LD      HL,0707H
OBF3 CD570C              CALL    RCBSET
OBF6 3E06                LD      A,6
OBF8 32EA08              LD      (RCB+10),A
OBFB CDF40C              CALL    BIOS1
OBFE 3AEC08              LD      A,(RCB+12)
OC01 C9                  RET
OC02                     ;
OC02 A7         B1SAVE:  AND     A
OC03 C9                  RET
OC04 3E03       B1LOAD:  LD      A,3
OC06 180A                JR      SETRFG
OC08 3E04       B2LOAD:  LD      A,4
OC0A 1806                JR      SETRFG
OC0C 3E02       SAVE:    LD      A,2
OC0E 1802                JR      SETRFG
OC10 3E05       VERIFY:  LD      A,5
OC12 32FD08     SETRFG:  LD      (RETFLG),A
OC15 AF                  XOR     A
OC16 32F010              LD      (10F0H),A
OC19 CDF40C              CALL    BIOS1
OC1C 3AF010              LD      A,(10F0H)
OC1F A7                  AND     A
OC20 C8                  RET     Z
OC21 37                  SCF
OC22 C9                  RET
OC23                     ;                     ODDC
OC23            ?DPCT1:  ENT
OC23 F5                  PUSH    AF
OC24 C5                  PUSH    BC
OC25 AF                  XOR     A
OC26 06C1                LD      B,C1H
OC28 B8                  CP      B
OC29 2002                JR      NZ,DP1
OC2B 3E1F                LD      A,1FH           ;CURSOR DOWN
OC2D 04         DP1:     INC     B
OC2E B8                  CP      B
OC2F 2002                JR      NZ,DP2
OC31 3E1E                LD      A,1EH           ;CURSOR UP
OC33 04         DP2:     INC     B
OC34 B8                  CP      B
OC35 2002                JR      NZ,DP3
OC37 3E1C                LD      A,1CH           ;CURSOR RIGHT
OC39 04         DP3:     INC     B
OC3A B8                  CP      B
OC3B 2002                JR      NZ,DP4
OC3D 3E1D                LD      A,1DH           ;CURSOR LEFT
OC3F 04         DP4:     INC     B
OC40 B8                  CP      B
OC41 2002                JR      NZ,DP5
OC43 3E0B                LD      A,0BH           ;CURSOR HOME
OC45 04         DP5:     INC     B
OC46 B8                  CP      B
OC47 2002                JR      NZ,DP6
OC49 3E0C                LD      A,0CH           ;WINDOW CLEAR
OC4B 04         DP6:     INC     B
OC4C B8                  CP      B
OC4D 2002                JR      NZ,DP7
OC4F 3E08                LD      A,08H           ;BACK SPACE (DEL)
OC51 CD8F0B     DP7:     CALL    PRNT
OC54 C1                  POP     BC
OC55 F1                  POP     AF
OC56 C9                  RET
OC57                     ;
OC57            RCBSET:  ENT
OC57 32E008              LD      (RCB),A         ;A=REQUEST NUMBER
OC5A E5                  PUSH    HL              ;H=OUTPUT BYTES
OC5B 2108E8              LD      HL,RCB8         ;L=INPUT BYTES
OC5E 22E208              LD      (RCB+2),HL
```

```
0C61 E1                 POP   HL
0C62 E5                 PUSH  HL
0C63 2E00               LD    L,0
0C65 22E408             LD    (RCB+4),HL
0C68 E1                 POP   HL
0C69 65                 LD    H,L
0C6A 2E00               LD    L,0
0C6C 22E608             LD    (RCB+6),HL
0C6F 65                 LD    H,L
0C70 22E808             LD    (RCB+8),HL
0C73 C9                 RET
0C74            ;
0C74            MOTON:  ENT
0C74 3EFF               LD    A,FFH
0C76 32E208             LD    (RCB+2),A
0C79 3E01               LD    A,1
0C7B 32E008             LD    (RCB),A
0C7E 1874               JR    BIOS1
0C80            ;
0C80            WIDTH1: ENT
0C80 E5                 PUSH  HL
0C81 3E10               LD    A,10H
0C83 32E008             LD    (RCB),A
0C86 21DD0B             LD    HL,CRDATA
0C89 7C                 LD    A,H
0C8A 65                 LD    H,L
0C8B 6F                 LD    L,A
0C8C 22E208             LD    (RCB+2),HL
0C8F 21000B             LD    HL,0B00H
0C92 22E408             LD    (RCB+4),HL
0C95 E1                 POP   HL
0C96 185C               JR    BIOS1
0C98            ;                             000C
0C98            PRNTS1: ENT
0C98 3E20               LD    A," "
0C9A 1802               JR    PRNT1
0C9C            ;                             003E
0C9C            BELL1:  ENT
0C9C 3E07               LD    A,7
0C9E            ;                             0012
0C9E            PRNT1:  ENT
0C9E FE0D               CP    0DH
0CA0 282F               JR    Z,LETNL1
0CA2 CD2F0D             CALL  ACCPRT
0CA5            ;
0CA5            STACUR: ENT
0CA5 E5                 PUSH  HL
0CA6 3E11               LD    A,11H
0CA8 210504             LD    HL,0405H
0CAB CD570C             CALL  RCBSET
0CAE 3E0A               LD    A,0AH
0CB0 32EA08             LD    (RCB+10),A
0CB3 CDF40C             CALL  BIOS1
0CB6 2AEB08             LD    HL,(RCB+11)
0CB9 227111             LD    (1171H),HL
0CBC E1                 POP   HL
0CBD C9                 RET
0CBE            ;                             0015
0CBE            MSG1:   ENT
0CBE D5                 PUSH  DE
0CBF 1A         MSG2:   LD    A,(DE)
0CC0 FE0D               CP    0DH
0CC2 2806               JR    Z,MSG3
0CC4 CD9E0C             CALL  PRNT1
0CC7 13                 INC   DE
0CC8 18F5               JR    MSG2
0CCA D1         MSG3:   POP   DE
0CCB C9                 RET
0CCC            ;                             0009
0CCC            NL1:    ENT
0CCC 3A7111             LD    A,(1171H)
0CCF A7                 AND   A
0CD0 C8                 RET   Z
0CD1            ;                             0006
0CD1            LETNL1: ENT
0CD1 3E0D               LD    A,0DH
0CD3 CD2F0D             CALL  ACCPRT
0CD6 3E0A               LD    A,0AH
0CD8 CD8F0B             CALL  PRNT
0CDB            ;
0CDB            ATBUTE: ENT
0CDB 3E10               LD    A,10H
0CDD E5                 PUSH  HL
0CDE 210006             LD    HL,0600H
0CE1 CD570C             CALL  RCBSET
0CE4 210302             LD    HL,0203H
0CE7 22EA08             LD    (RCB+10),HL
0CEA 3AFC08             LD    A,(ATCODE)
0CED 67                 LD    H,A
0CEE 2E11               LD    L,11H
0CF0 22EC08             LD    (RCB+12),HL
0CF3 E1                 POP   HL
0CF4            ;
0CF4            BIOS1:  ENT
0CF4 AF                 XOR   A
0CF5 3205FD             LD    (FD05H),A
0CF8 00                 NOP
```

Z80 システム・モニタ アセンブル・リスト

```
0CF9 C9                       RET
0CFA                ;                      001E
0CFA                BRKEY1:  ENT
0CFA CD470D                   CALL   GETKY1
0CFD FE03                     CP     3
0CFF C9                       RET
0D00                ;                      09B3
0D00                ??KEY1:  ENT
0D00 CD0E0D                   CALL   CURON
0D03 CD470D         KEY1:     CALL   GETKY1
0D06 A7                       AND    A
0D07 28FA                     JR     Z,KEY1
0D09                ;
0D09                CUROFF:  ENT
0D09 F5                       PUSH   AF
0D0A 3E16                     LD     A,16H
0D0C 1803                     JR     CUR1
0D0E                ;
0D0E                CURON:   ENT
0D0E F5                       PUSH   AF
0D0F 3E17                     LD     A,17H
0D11 32EB08         CUR1:     LD     (RCB+11),A
0D14 E5                       PUSH   HL
0D15 210004                   LD     HL,0400H
0D18 3E10                     LD     A,10H
0D1A CD570C                   CALL   RCBSET
0D1D 3E0C                     LD     A,0CH
0D1F 32EA08                   LD     (RCB+10),A
0D22 CDF40C                   CALL   BIOS1
0D25 E1                       POP    HL
0D26 F1                       POP    AF
0D27 C9                       RET
0D28                ;
0D28                LPRNT1:  ENT
0D28 32E808                   LD     (RCB+8),A
0D2B 3E0E                     LD     A,0EH
0D2D 1805                     JR     PRT1
0D2F                ;
0D2F                ACCPRT:  ENT
0D2F 32E808                   LD     (RCB+8),A
0D32 3E14                     LD     A,14H
0D34 32E008         PRT1:     LD     (RCB),A
0D37 E5                       PUSH   HL
0D38 210001                   LD     HL,0100H
0D3B 22E408                   LD     (RCB+4),HL
0D3E 2108E8                   LD     HL,RCB8
0D41 22E208                   LD     (RCB+2),HL
0D44 E1                       POP    HL
0D45 18AD                     JR     BIOS1
0D47                ;                      001B
0D47                GETKY1:  ENT
0D47 3E15                     LD     A,15H
0D49 E5                       PUSH   HL
0D4A 210001                   LD     HL,0100H
0D4D CD570C                   CALL   RCBSET
0D50 E1                       POP    HL
0D51 CDF40C                   CALL   BIOS1
0D54 3AE808                   LD     A,(RCB+8)
0D57 C9                       RET
0D58                ;
0D58                GETCA:   ENT
0D58 F5                       PUSH   AF
0D59 3E13                     LD     A,13H
0D5B 1803                     JR     GETO
0D5D                ;
0D5D                GET:     ENT
0D5D F5                       PUSH   AF
0D5E 3E12                     LD     A,12H
0D60 E5             GETO:     PUSH   HL
0D61 D5                       PUSH   DE
0D62 21FF00                   LD     HL,00FFH
0D65 CD570C                   CALL   RCBSET
0D68 21043D                   LD     HL,BUF
0D6B 22E208                   LD     (RCB+2),HL
0D6E CDF40C                   CALL   BIOS1
0D71 3A3E04                   LD     A,(BUFF+1)
0D74 FE07                     CP     7
0D76 2813                     JR     Z,GET2
0D78 213D04                   LD     HL,BUFF
0D7B 3AE508                   LD     A,(RCB+5)
0D7E 1600                     LD     D,0
0D80 5F                       LD     E,A
0D81 19                       ADD    HL,DE
0D82 360D                     LD     (HL),0DH
0D84 3A3E04                   LD     A,(BUFF+1)
0D87 FE0D                     CP     0DH
0D89 2005                     JR     NZ,GET3
0D8B 3E1B           GET2:     LD     A,1BH
0D8D 324104                   LD     (BUFF+4),A
0D90 D1             GET3:     POP    DE
0D91 E1                       POP    HL
0D92 F1                       POP    AF
0D93 C9                       RET
0D94                ;                      0003
0D94                GETL1:   ENT
0D94 CD0E0D                   CALL   CURON
0D97 CD5D0D                   CALL   GET
0D9A CDA80D                   CALL   GETSUB
```

```
OD9D F5                    PUSH  AF
OD9E CD860B                CALL  LETNL
ODA1 F1                    POP   AF
ODA2 C3090D                JP    CUROFF
ODA5              ;
ODA5              GETC1:   ENT
ODA5 CD580D                CALL  GETCA
ODA8              ;
ODA8              GETSUB:  ENT
ODA8 F5                    PUSH  AF
ODA9 E5                    PUSH  HL
ODAA D5                    PUSH  DE
ODAB C5                    PUSH  BC
ODAC 214104                LD    HL,BUFF+4
ODAF EB                    EX    DE,HL
ODB0 064F                  LD    B,79
ODB2 1A           GETS1:   LD    A,(DE)
ODB3 77                    LD    (HL),A
ODB4 FE1B                  CP    1BH
ODB6 280A                  JR    Z,GETS2
ODB8 FE0D                  CP    ODH
ODBA 2806                  JR    Z,GETS2
ODBC 13                    INC   DE
ODBD 23                    INC   HL
ODBE 10F2                  DJNZ  GETS1
ODC0 360D                  LD    (HL),ODH
ODC2 C1           GETS2:   POP   BC
ODC3 D1                    POP   DE
ODC4 E1                    POP   HL
ODC5 F1                    POP   AF
ODC6 C9                    RET
ODC7              ;                             03BA
ODC7              H4PR1:   ENT
ODC7 7C                    LD    A,H
ODC8 CDB00B                CALL  HEX2PR
ODCB 7D                    LD    A,L
ODCC              ;                             03C3
ODCC              H2PR1:   ENT
ODCC F5                    PUSH  AF
ODCD E6F0                  AND   F0H
ODCF 0F                    RRCA
ODD0 0F                    RRCA
ODD1 0F                    RRCA
ODD2 0F                    RRCA
ODD3 CDB30B                CALL  ASC
ODD6 CD8F0B                CALL  PRNT
ODD9 F1                    POP   AF
ODDA E60F                  AND   0FH
ODDC CDB30B                CALL  ASC
ODDF C38F0B                JP    PRNT
ODE2              ;                             03DA
ODE2              ASC1:    ENT
ODE2 FE0A                  CP    0AH
ODE4 3802                  JR    C,ASC2
ODE6 C607                  ADD   A,7
ODE8 C630         ASC2:    ADD   A,30H
ODEA C9                    RET
ODEB              ;                             03F9
ODEB              HEX1:    ENT
ODEB FE30                  CP    30H
ODED D8                    RET   C
ODEE FE3A                  CP    3AH
ODF0 3809                  JR    C,HEXA
ODF2 FE41                  CP    41H
ODF4 D8                    RET   C
ODF5 FE47                  CP    47H
ODF7 3005                  JR    NC,HEXB
ODF9 D607                  SUB   7
ODFB D630         HEXA:    SUB   30H
ODFD C9                    RET
ODFE 37           HEXB:    SCF
ODFF C9                    RET
0E00              ;                             0410
0E00              HLHEX1:  ENT
0E00 D5                    PUSH  DE
0E01 CDBC0B                CALL  2HEX
0E04 3807                  JR    C,HLHEX2
0E06 67                    LD    H,A
0E07 CDBC0B                CALL  2HEX
0E0A 3801                  JR    C,HLHEX2
0E0C 6F                    LD    L,A
0E0D D1           HLHEX2:  POP   DE
0E0E C9                    RET
0E0F              ;                             041F
0E0F              2HEX1:   ENT
0E0F C5                    PUSH  BC
0E10 1A                    LD    A,(DE)
0E11 13                    INC   DE
0E12 CDB60B                CALL  HEX
0E15 380D                  JR    C,2HEX2
0E17 07                    RLCA
0E18 07                    RLCA
0E19 07                    RLCA
0E1A 07                    RLCA
0E1B 4F                    LD    C,A
0E1C 1A                    LD    A,(DE)
0E1D 13                    INC   DE
0E1E CDB60B                CALL  HEX
```

Z80 システム・モニタ アセンブル・リスト

```
0E21 3801              JR    C,2HEX2
0E23 B1                OR    C
0E24 C1        2HEX2:  POP   BC
0E25 C9                RET
0E26           ;
0E26           RET1:   ENT
0E26 3E01              LD    A,1
0E28 32FD08            LD    (RETFLG),A
0E2B CDF40C            CALL  BIOS1
0E2E           ;
0E2E                   SKP   H
0E2E           ;**********************************
0E2E           ;*  Z-80 Tiny System Monitor      *
0E2E           ;* Copyright (C) 1982             *
0E2E           ;*              BY ELECOM Soft    *
0E2E           ;*   Written by Jun.Amanai        *
0E2E           ;**********************************
0E2E           ;
0E2E           MAIN1:  ENT                   ;INITIALIZE
0E2E 31F010            LD    SP,10F0H
0E31 3E07              LD    A,7             ;WHITE COLOR
0E33 32FC08            LD    (ATCODE),A
0E36 CDD10B            CALL  WIDTH           ;WIDTH SET
0E39 11C30F            LD    DE,TITLE        ;TITLE PRINT
0E3C CD920B            CALL  MSG
0E3F           COMAND: ENT
0E3F 31F010            LD    SP,10F0H
0E42 CD860B            CALL  LETNL
0E45 3E3E              LD    A,">"
0E47 CD8F0B            CALL  PRNT
0E4A 11A311    SCCOM:  LD    DE,11A3H        ;FOR SCREEN EDIT
0E4D CD830B            CALL  GETL
0E50 1A                LD    A,(DE)
0E51 FE3E              CP    ">"             ;SCREEN EDIT
0E53 C2BA0E            JP    NZ,EDIT
0E56 13                INC   DE
0E57 1A                LD    A,(DE)
0E58 13                INC   DE
0E59 13                INC   DE
0E5A F5                PUSH  AF
0E5B CDB90B            CALL  HLHEX
0E5E F1                POP   AF
0E5F           ;
0E5F           ;       RETURN TO F-BASIC
0E5F           ;         FORMAT  >!
0E5F           ;
0E5F FE21              CP    "!"
0E61 28C3              JR    Z,RET1
0E63           ;
0E63           ;       CMT MOTOR ON
0E63           ;         FOMAT  >@
0E63           ;
0E63 FE40              CP    "@"
0E65 2005              JR    NZ,COM1
0E67 CD740C            CALL  MOTON
0E6A 18D3              JR    COMAND
0E6C           ;
0E6C           ;       GO TO NEW LOCATION
0E6C           ;         FORMAT  >G B000
0E6C           ;
0E6C FE47      COM1:   CP    "G"
0E6E 2001              JR    NZ,COM1A
0E70 E9                JP    (HL)
0E71 FE4A      COM1A:  CP    "J"
0E73 2004              JR    NZ,COM2
0E75 2A0611            LD    HL,(1106H)
0E78 E9                JP    (HL)
0E79 FE4D      COM2:   CP    "M"             ;MEMORY CHANGE
0E7B 2860              JR    Z,CHANGE
0E7D FE57              CP    "W"             ;PROGRAM SAVES TO TAPE
0E7F CA300F            JP    Z,WRITE
0E82 FE4C              CP    "L"             ;LOAD FROM CMT
0E84 CA7A0F            JP    Z,LOAD
0E87 FE44              CP    "D"             ;MEMORY DUMP
0E89 20B4              JR    NZ,COMAND
0E8B           ;
0E8B           ;      MEMORY DUMP & CHECKSUM
0E8B           ;        FORMAT  >D 2000
0E8B           ;
0E8B           DUMP:   ENT
0E8B E5                PUSH  HL
0E8C CD860B            CALL  LETNL
0E8F CDAD0B            CALL  HEX4PR
0E92 0608              LD    B,8
0E94 08                EX    AF,AF'
0E95 AF                XOR   A
0E96 08                EX    AF,AF'
0E97 CD8C0B    LINE1:  CALL  PRNTS
0E9A 7E                LD    A,(HL)
0E9B 08                EX    AF,AF'
0E9C 86                ADD   A,(HL)
0E9D 08                EX    AF,AF'
0E9E CDB00B            CALL  HEX2PR
0EA1 23                INC   HL
0EA2 10F3              DJNZ  LINE1
0EA4 CD8C0B            CALL  PRNTS
0EA7 3E3A              LD    A,3AH           ; ":"
0EA9 CD8F0B            CALL  PRNT
0EAC 08                EX    AF,AF'
```

```
OEAD CDB00B              CALL  HEX2PR
OEB0 E1                  POP   HL
OEB1 110800              LD    DE,8
OEB4 19                  ADD   HL,DE
OEB5 CDB40F              CALL  PAUSE
OEB8 18D1                JR    DUMP
OEBA                 ;
OEBA                 ;    SCREEN EDIT
OEBA                 ;
OEBA                 EDIT:  ENT
OEBA FE1B                CP    1BH                      ;EDITING COMPLETE ?
OEBC CA3F0E          ZCOM:  JP    Z,COMAND
OEBF CDB90B              CALL  HLHEX
OEC2 DA3F0E          CYCOM: JP    C,COMAND
OEC5 11AB11              LD    DE,11ABH
OEC8 0608                LD    B,8                      ;8 BYTES
OECA CDBC0B          EDIT1: CALL  2HEX
OECD 38F3                JR    C,CYCOM
OECF 77                  LD    (HL),A
OED0 23                  INC   HL
OED1 13                  INC   DE
OED2 10F6                DJNZ  EDIT1
OED4 11A311              LD    DE,11A3H
OED7 CDC80B              CALL  GETC
OEDA 1A                  LD    A,(DE)
OEDB 18DD                JR    EDIT
OEDD                 ;
OEDD                 ;    MEMORY CHANGE
OEDD                 ;       FORMAT  >M AB00
OEDD                 ;
OEDD                 CHANGE: ENT
OEDD 11A311              LD    DE,11A3H
OEE0 CDAD0B              CALL  HEX4PR
OEE3 CD8C0B              CALL  PRNTS
OEE6 7E                  LD    A,(HL)
OEE7 CDB00B              CALL  HEX2PR
OEEA 3E2D                LD    A,2DH                    ; "-"
OEEC CD8F0B              CALL  PRNT
OEEF CDC20B          ERRBCK: CALL  ??KEY
OEF2 FE20                CP    20H
OEF4 2831                JR    Z,BACK
OEF6 FE0D                CP    0DH
OEF8 2830                JR    Z,FWD
OEFA FE03                CP    3
OEFC 28BE                JR    Z,ZCOM
OEFE 12                  LD    (DE),A
OEFF CD8F0B              CALL  PRNT
OF02 CDC20B              CALL  ??KEY
OF05 13                  INC   DE
OF06 12                  LD    (DE),A
OF07 CD8F0B              CALL  PRNT
OF0A 1B                  DEC   DE
OF0B CDBC0B              CALL  2HEX
OF0E 3807                JR    C,ERR
OF10 77                  LD    (HL),A
OF11 23                  INC   HL
OF12 CD860B              CALL  LETNL
OF15 18C6                JR    CHANGE
OF17 CDAA0B          ERR:   CALL  BELL
OF1A CD8C0B              CALL  PRNTS
OF1D 3E3F                LD    A,"?"
OF1F CD8F0B              CALL  PRNT
OF22 11A311              LD    DE,11A3H
OF25 18C8                JR    ERRBCK
OF27 2B              BACK:  DEC   HL
OF28 1801                JR    FWD1
OF2A 23              FWD:   INC   HL
OF2B CD860B          FWD1:  CALL  LETNL
OF2E 18AD                JR    CHANGE
OF30                 ;
OF30                 ;    WRITE TO CASSETE TAPE
OF30                 ;       FORMAT  >W 1A00,2FFF,2000,HUDSON
OF30                 ;
OF30                 WRITE: ENT
OF30 220411              LD    (1104H),HL
OF33 E5                  PUSH  HL
OF34 11AB11              LD    DE,11ABH
OF37 CDB90B              CALL  HLHEX
OF3A D1                  POP   DE
OF3B ED52                SBC   HL,DE
OF3D 23                  INC   HL
OF3E 220211              LD    (1102H),HL
OF41 11B011              LD    DE,11B0H
OF44 CDB90B              CALL  HLHEX
OF47 220611              LD    (1106H),HL
OF4A 0608                LD    B,8
OF4C 11B511              LD    DE,11B5H
OF4F 21F110              LD    HL,10F1H
OF52 1A              NAMEIN: LD    A,(DE)
OF53 FE0D                CP    0DH
OF55 2807                JR    Z,WSPC
OF57 77                  LD    (HL),A
OF58 23                  INC   HL
OF59 13                  INC   DE
OF5A 10F6                DJNZ  NAMEIN
OF5C 1805                JR    CMPLNM
OF5E 3620            WSPC:  LD    (HL),20H
OF60 23                  INC   HL
OF61 10FB                DJNZ  WSPC
```

Z80 システム・モニタ アセンブル・リスト

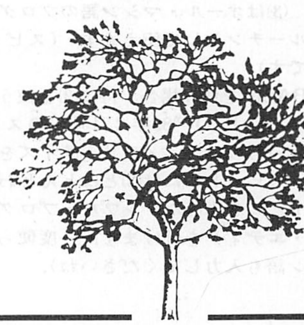
```
0F63 11EF0F        CMPLNM: LD    DE,WRITNG
0F66 CD920B                CALL  MSG
0F69 CD0C0C                CALL  SAVE
0F6C 3009          TAPEOP: JR    NC,AYCOM
0F6E CD860B                CALL  LETNL
0F71 11F70F                LD    DE,ERRMSG
0F74 CD920B                CALL  MSG
0F77 C33F0E        AYCOM:  JP    COMAND
0F7A                       ;
0F7A                       ;  LOAD PROGRAM
0F7A                       ;    FORMAT  >L
0F7A                       ;
0F7A               LOAD:   ENT
0F7A CD040C                CALL  B1LOAD
0F7D 38ED                  JR    C,TAPEOP
0F7F 11D70F                LD    DE,FOUND
0F82 CD920B                CALL  MSG
0F85 11F110                LD    DE,10F1H
0F88 CD920B                CALL  MSG
0F8B CD860B                CALL  LETNL
0F8E 11DF0F                LD    DE,LOAD?
0F91 CD920B                CALL  MSG
0F94 CDC20B                CALL  ??KEY
0F97 F5                    PUSH  AF
0F98 CD8F0B                CALL  PRNT
0F9B F1                    POP   AF
0F9C FE59                  CP    "Y"
0F9E 2806                  JR    Z,MEMIN
0FA0 FE53                  CP    "S"
0FA2 28D6                  JR    Z,LOAD
0FA4 18D1                  JR    AYCOM
0FA6 CD860B        MEMIN:  CALL  LETNL
0FA9 11E70F                LD    DE,LOADIN
0FAC CD920B                CALL  MSG
0FAF CD080C                CALL  B2LOAD
0FB2 18B8                  JR    TAPEOP
0FB4                       ;
0FB4                       ;  PAUSE & BREAK CHECK
0FB4                       ;
0FB4               PAUSE:  ENT
0FB4 CD950B                CALL  GETKY
0FB7 FE1B                  CP    1BH
0FB9 2003                  JR    NZ,BREAK?
0FBB CDC20B                CALL  ??KEY
0FBE FE03          BREAK?: CP    3
0FC0 C0                    RET   NZ
0FC1 18B4                  JR    AYCOM
0FC3                       ;
0FC3                       ;
0FC3 5A2D3830      TITLE:  DEFM  'Z-80 SYSTEM MONITOR'
0FC7 20535953
0FCB 54454D20
0FCF 4D4F4E49
0FD3 544F52
0FD6 0D                    DEFB  0DH
0FD7                       ;
0FD7 466F756E      FOUND:  DEFM  'Found: '
0FDB 643A20
0FDE 0D                    DEFB  0DH
0FDF                       ;
0FDF 4C6F6164      LOAD?:  DEFM  'Load ? '
0FE3 203F20
0FE6 0D                    DEFB  0DH
0FE7                       ;
0FE7 4C6F6164      LOADIN: DEFM  'Loading'
0FEB 696E67
0FEE 0D                    DEFB  0DH
0FEF                       ;
0FEF 57726974      WRITNG: DEFM  'Writing'
0FF3 696E67
0FF6 0D                    DEFB  0DH
0FF7                       ;
0FF7 07            ERRMSG: DEFB  7
0FF8 4572726F              DEFM  'Error'
0FFC 72
0FFD 0D                    DEFB  0DH
0FFE                       ;
0FFE                       END
```

Z80 システム・モニタ ラベル・テーブル

2HEX	0BBC	2HEX1	0E0F	2HEX2	0E24	??KEY	0BC2	??KEY1	0D00
?DPCT	0BBF	?DPCT1	0C23	?RD1	0BA1	?RDD	0BA4	?VRFY	0BA7
?WR1	0B9B	?WRD	0B9E	ACCPRT	0D2F	ASC	0BB3	ASC1	0DE2
ASC2	0DE8	ATBUTE	0CDB	ATCODE	08FC	AUX1	0BD7	AUX2	0BDA
AYCOM	0F77	B1LOAD	0C04	B1SAVE	0C02	B2LOAD	0C08	BACK	0F27
BELL	0BAA	BELL1	0C9C	BIOS	0BCB	BIOS1	0CF4	BREAK?	0FBE
BRKEY	0B98	BRKEY1	0CFA	BUF	3D04	BUFF	043D	CHANGE	0EDD
CMPLNM	0F63	COM1	0E6C	COM1A	0E71	COM2	0E79	COMAND	0E3F
CRDATA	0BDD	CUR1	0D11	CUROFF	0D09	CURON	0D0E	CYCOM	0EC2
DP1	0C2D	DP2	0C33	DP3	0C39	DP4	0C3F	DP5	0C45
DP6	0C4B	DP7	0C51	DUMP	0E8B	EDIT	0EBA	EDIT1	0ECA
ERR	0F17	ERRBCK	0EEF	ERRMSG	0FF7	FOUND	0FD7	FWD	0F2A
FWD1	0F2B	GET	0D5D	GET0	0D60	GET2	0D8B	GET3	0D90
GETC	0BC8	GETC1	0DA5	GETCA	0D58	GETKY	0B95	GETKY1	0D47

Z80　システム・モニタ　ラベル・テーブル

GETL	0B83	GETL1	0D94	GETS1	0DB2	GETS2	0DC2	GETSUB	0DA8
H2PR1	0DCC	H4PR1	0DC7	HEX	0BB6	HEX1	0DEB	HEX2PR	0BB0
HEX4PR	0BAD	HEXA	0DFB	HEXB	0DFE	HLHEX	0BB9	HLHEX1	0E00
HLHEX2	0E0D	KEY1	0D03	LETNL	0B86	LETNL1	0CD1	LINE1	0E97
LOAD	0F7A	LOAD?	0FDF	LOADIN	0FE7	LPRNT	0BD4	LPRNT1	0D28
MAIN	0B80	MAIN1	0E2E	MEMIN	0FA6	MOTON	0C74	MSG	0B92
MSG1	0CBE	MSG2	0CBF	MSG3	0CCA	NAMEIN	0F52	NL	0B89
NL1	0CCC	PAUSE	0FB4	PRNT	0B8F	PRNT1	0C9E	PRNTS	0B8C
PRNTS1	0C98	PRT1	0D34	RCB	0BE0	RCB8	E808	RCBSET	0C57
RET1	0E26	RETFLG	0BFD	RETURN	0BC5	SAVE	0C0C	SCCOM	0E4A
SCREEN	0BCE	SCRN	0BE8	SETRFG	0C12	STACUR	0CA5	TAPEOP	0F6C
TITLE	0FC3	VERIFY	0C10	WDATA	0BE1	WIDTH	0BD1	WIDTH1	0CB0
WRITE	0F30	WRITNG	0FEF	WSPC	0F5E	ZCOM	0EBC		



マシン語データ↔配列変換ルーチン

■馬場二郎

BASICでグラフィック・データをPUT文で扱うとき，マシン語データまたは，DATA*を配列に落とすのに非常な時間をかけていますが，この作業は機械語レベルで考えると恐ろしく無駄なことをやっています．

サブルーチンは3つあり，それぞれに機械語があります．(1)はGET文で読んだ配列（整数型に限ります）をマシン語データとしてメモリに落とすもの．(2)はその逆にPUT文で用いるために配列に格納するもの．(3)は配列を介さず直接GET文などで作成したマシン語データをPUTするものです（この場合はグラフィック・データに限ります）．

動作原理

(1)：配列→マシン語データ

(2)：配列←マシン語

(3)：マシン語データPUT

(1)，(2)の動作原理は簡単で，BASICの整数型配列が，配列数（2バイト）＋要素となっているので，単にメモリ・ブロックの転送にすぎません（文字，カラー対応のいかんを問いません）．(3)はBASICのPUT文ルーチンを借用しています．

以上非常に簡単なものですが，実用的には，

❶ PUT文用の配列へデータを格納する待ち時間がなくなる（スピードはBASICとは問題外です）．

❷ 同一配列を複数のグラフィック・データ用として使うことができ，メモリの節約になる．

❸ PUT文の読み込み時間がなくなるので実行中ストップしても気にならない．

❺ (3)はオール・マシン語のプログラムでも表示用のサブルーチンとして使えます（スピードもかなり速いものです）．

BASICを使う場合，特に(3)を使う必要性はほとんどありません．I/O〝82年2月号〟のスタートレックのような大きいものでは，メモリ・サイズを節約する目的に使えます．スピード的には(2)とほとんどわからないと思います．

(1)と(2)を使ったサンプル・プログラムとしてグラフィック・エディタを付けます．一度使ってみてください（マシン語も入力してくださいね）．

最後に

BASICについては最近いろいろ出ているのであえて書きませんが，1つ，PUTなどはデータが連続していますがサブシステムとの共有エリアは128バイトしかなく，PUT文に相当するサブコマンドだけで制御することは不可能です．実は継続データを示すサブコマンドが他にあります．

$80	$	64 継続データ
0	1	2

これはGET文やPUT文のすべてに共通です．
（くわしくは$EB86を逆アセンブル）

まだ継続するときは$80？

◘参考文献

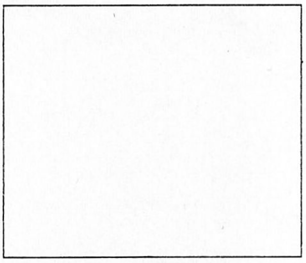
1）インターフェース，'81年11月，'82年2月号

2）I/O，'82年12月号

3）その他雑多

X-Size?300
Y-Size?150

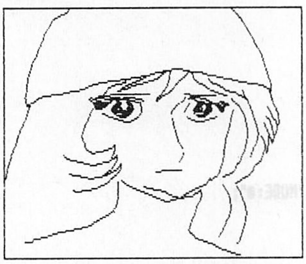
Size=&H15FA ﾊﾞｲﾄ



RUNするとX方向とY方向のドット・サイズを聞いてくるので，任意の値を入れます．（X＝1～538，Y＝1～168）．
指定サイズの枠がかかれると同時に配列のサイズが右上に表示されます．

X-Size?300
Y-Size?150

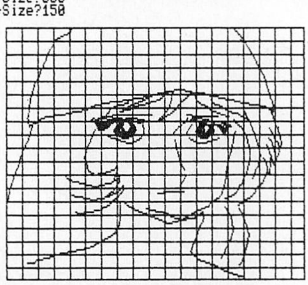
Size=&H15FA ﾊﾞｲﾄ



エディタを使って絵をプロットしてゆきます．このモードではメモリをほとんど使用せず，またエディタの速度も早いですが，万一［STOP］1を押したときのためにこまめに“S”コマンドでデータをメモリ上にセーブしてください．

X-Size?300
Y-Size?150

Size=&H15FA ﾊﾞｲﾄ



“Q”コマンドで網目を出力します．このモードでは画面とともに対応するメモリ上のワーク・エリアにも同時にドットを更新します．操作上は“S”コマンドが実際上使えないこと以外は変わりませんが，若干速度は遅くなります．私のように絵心のないものでもこれなら下書きの絵をうまく写すことわできます．
また，このモードなら途中STOPしてもRUNして，“L”コマンドでワーク・エリアを戻してやることができ安心です．

X-Size?300
Y-Size?150

Size=&H15FA ﾊﾞｲﾄ



“R”コマンドで網目だけを消します．
（実際には画面をクリアして，メモリ上のマシン語データを配列に落しPUTするのですがPUTの時間は2～3秒しかかかりません）
画面のモードは前に戻ります．

MODE:0

テン・キー

↖	↑	↗
←	・	→
↙	↓	↘

［+］現在のカーソルの位置を知らないとき，カーソルを他の場所へ移したいときに使います．グラフィック・カーソルが出るので，動かして［RETURN］してください．

［・］カーソル位置のドットをプロットするか消去するかのモードを切り換える．

［P］現在のカーソル位置をセーブする．

［H］Pでセーブした位置へカーソルを移す．
（分枝×に便利です．）

［S］画面データをメモリ上にマシン語データとしてセーブします．セーブするアドレスを聞いてくるので，&Hなしの16進で入力してください．このときエリアが重ならないように注意してください．

［L］Sでセーブしたデータを画面上に出します．このときもアドレスを聞いてくるので入力してください．

［C］画面をクリアする．念のため再度OKか聞いてきます．

［Q］8×16ドットの網目を出力とともにメモリのワークエリアも同時に更新するモードになります．Sと同様にワーク・アドレスを入力してください．

［R］網目だけを消します．モードは元に戻ります．

［E］STOP

マシン語データ↔配列変換サブルーチン

```
                            ****************************
                            * ｷｶｲｺﾞ ｻﾌﾞﾙｰﾁﾝ            *
                            ****************************
      7500                         ORG    $7500
      7500         0000     ADDR   FDB    0
                            * ﾊｲﾚﾂ -> ﾏｼﾝｺﾞﾃﾞｰﾀ
      7502 34      72       GETSAV PSHS   X,Y,U,A
      7504 EE      02              LDU    2,X
      7506 10AE    5E              LDY    -2,U
      7509 AE      8D FFF3         LDX    >ADDR,PCR
      750D EC      C1       LOOP1  LDD    ,U++
      750F ED      81              STD    ,X++
      7511 31      3F              LEAY   -1,Y
      7513 26      F8    750D      BNE    LOOP1
      7515 35      F2              PULS   X,Y,U,A,PC
                            *
                            *ﾊｲﾚﾂ <- ﾏｼﾝｺﾞﾃﾞｰﾀ
      7517 34      72       LOAD   PSHS   X,Y,U,A
      7519 AE      02              LDX    2,X
      751B 10AE    1E              LDY    -2,X
      751E EE      8D FFDE         LDU    >ADDR,PCR
      7522 20      E9    750D      BRA    LOOP1
                            *
                            *
                            *ﾏｼﾝｺﾞﾃﾞｰﾀ PUT
      7524 34      7E              PSHS   A,B,X,Y,U,DP
      7526 AE      02              LDX    2,X
      7528 EC      81              LDD    ,X++
      752A FD      00C5            STD    >$00C5
      752D EC      81              LDD    ,X++
      752F FD      00C7            STD    >$00C7
      7532 EC      81              LDD    ,X++
      7534 FD      00C9            STD    >$00C9
      7537 EC      81              LDD    ,X++
      7539 FD      00CB            STD    >$00CB
      753C EC      81              LDD    ,X++
      753E FD      00F3            STD    >$00F3
      7541 EC      81              LDD    ,X++
      7543 FD      00F5            STD    >$00F5
      7546 EC      81              LDD    ,X++
      7548 F7      01A5            STB    >$01A5
      754B EC      81              LDD    ,X++
      754D F7      00C2            STB    >$00C2
      7550 EC      84              LDD    ,X
      7552 F7      00FB            STB    >$00FB
      7555 BD      EB27            JSR    $EB27   ;BASIC ｻﾌﾞﾙｰﾁﾝ
      7558 35      FE              PULS   A,B,X,Y,U,DP,PC
                                   END
TOTAL ERRORS 00000--00000
TOTAL WARNINGS 00000--00000

PROGRAM BEGIN ADDR=7500
PROGRAM END   ADDR=7559
PROGRAM ENTRY ADDR=****
```

グラフィック・パターン　作成プログラム

```
1 '*******************************************************
2 '* ｸﾞﾗﾌｨｯｸ ﾊﾟﾀｰﾝ ｻｸｾｲ ﾌﾟﾛｸﾞﾗﾑ                            *
3 '*******************************************************
4 '
10 WIDTH80,25:DEF USR0=&H7502:DEF USR1=&H7517
20 LOCATE 10,1:LINE INPUT "X-Size?";S$:SX=VAL(S$):IF (SX>538)OR(SX<1)THEN 20
30 LOCATE 10,2:LINE INPUT "Y-Size?";S$:SY=VAL(S$):IF (SY>168)OR(SY<1)THEN 30
40 SIZE=INT(SX*SY/16)+1:LOCATE 60,2:PRINT "Size=&H";HEX$(SIZE*2);" ﾊﾞｲﾄ"
50 DIM A%(SIZE)
60 LINE@ (100,30)-(101+SX,31+SY),PSET,,B
70 SW=1:X=INT(100+SX/2):Y=INT(30+SY/2):XP=X:YP=Y:SW2=0
75 '
80 '***MAIN*************************************************
90 K$=INKEY$:IF K$="" THEN 90
100 IF K$="9" THEN IF (X<100+SX) AND (Y>31 ) THEN X=X+1:Y=Y-1:GOSUB280
110 IF K$="8" THEN IF                (Y>31 ) THEN       Y=Y-1:GOSUB280
120 IF K$="7" THEN IF (X>101 ) AND (Y>31 ) THEN X=X-1:Y=Y-1:GOSUB280
130 IF K$="6" THEN IF (X<100+SX)                THEN X=X+1 :GOSUB280
140 IF K$="5" THEN                                           GOSUB280
150 IF K$="4" THEN IF (X>101 )                 THEN X=X-1:GOSUB280
160 IF K$="3" THEN IF (X<100+SX) AND (Y<30+SY) THEN X=X+1:Y=Y+1:GOSUB280
170 IF K$="2" THEN IF                (Y<30+SY) THEN       Y=Y+1:GOSUB280
180 IF K$="1" THEN IF (X>101 ) AND (Y<30+SY) THEN X=X-1:Y=Y-1:GOSUB280
190 IF K$="." THEN IF SW=1 THEN SW=0:LOCATE 60,10:PRINT "MODE:○"; ELSE SW=1:LOCATE 60,10:PRINT "MODE:●";
200 IF K$="+" THEN GCURSOR (X,Y),(X,Y)
210 IF K$="P" THEN XP=X:YP=Y
220 IF K$="H" THEN X=XP:Y=YP
230 IF K$="L" THEN GOSUB370
240 IF K$="S" THEN GOSUB310
250 IF K$="C" THEN LOCATE 60,12:LINE INPUT "CLMAR OK?(Y)";S$:IF S$="Y" THEN LINE@ (101,31)-(100+SX,30+SY),PRESET,,BF:LOCATE 60,12:PR
INT SPACE$(15); ELSE LOCATE 60,12:PRINT SPACE$(15)
252 IF K$="Q" THEN IF SW2=0 THEN SW2=1:GOSUB1000
254 IF K$="R" THEN IF SW2=1 THEN SW2=0:GOSUB2000
260 IF K$="E" THEN STOP
270 GOTO 80
280 IF SW=1 THEN PSET(X,Y) ELSE PRESET(X,Y)
290 IF SW2=0 THEN RETURN
292 ADD=INT(((Y-31)*SX+X-101)/8)+WORK
293 SI=(((Y-31) MOD 8)*SX+X-101) MOD 8:BIT=2^(7-SI)
294 W=PEEK(ADD):IF SW=1 THEN POKE ADD,(BIT OR W) ELSE POKE ADD,(W AND (NOT BIT))
300 RETURN
310 '***SAVE************************************************
320 GET@ (101,31)-(100+SX,30+SY),A%,G
330 LOCATE 60,20:LINE INPUT"SAVE ADDRESS?";S$:W=VAL("&H"+S$):IF S$="" THEN 360
340 POKE &H7500,INT(W/256):POKE &H7501,(W MOD 256)
350 W=USR0(VARPTR(A%(0)))
360 BEEP1:FOR I=1 TO 100:NEXT:BEEP0:LOCATE 60,12:PRINT SPACE$(19);:RETURN
370 '***LOAD************************************************
390 LOCATE 60,21:LINE INPUT"LOAD ADDRESS?";S$:W=VAL("&H"+S$):IF S$="" THEN 430
400 POKE &H7500,INT(W/256):POKE &H7501,(W MOD 256)
410 W=USR1(VARPTR(A%(0)))
415 LINE@ (101,31)-(100+SX,30+SY),PRESET,,BF
420 PUT@ (101,31)-(100+SX,30+SY),A%,PSET
430 BEEP1:FOR I=1 TO 100:NEXT:BEEP0:LOCATE 60,21:PRINT SPACE$(19);:RETURN
1000 '***CURSOR*********************************************
1010 GET@ (101,31)-(100+SX,30+SY),A%,G
1020 LOCATE 60,20:LINE INPUT"WORK ADDRESS?";S$:W=VAL("&H"+S$):IF S$="" THEN 360
1030 WORK=W:POKE &H7500,INT(W/256):POKE &H7501,(W MOD 256)
1040 W=USR0(VARPTR(A%(0)))
1042 FOR I=38 TO 30+SY STEP 8:LINE@ (100,I)-(100+SX,I),PSET:NEXT
1044 FOR I=116TO100+SX STEP16:LINE@ (I,30)-(I,30+SY),PSET:NEXT
1050 BEEP1:FOR I=1 TO 100:NEXT:BEEP0:LOCATE 60,20:PRINT SPACE$(19);:RETURN
2000 '***CURSOR CLEAR***************************************
2030 W=USR1(VARPTR(A%(0)))
2040 LINE@ (101,31)-(100+SX,30+SY),PRESET,,BF
2050 PUT@ (101,31)-(100+SX,30+SY),A%,PSET
2060 BEEP1:FOR I=1 TO 100:NEXT:BEEP0:LOCATE 60,21:PRINT SPACE$(19);:RETURN
```

配列↔マシン語変換サブルーチン

```
100 '**************************************************************************
110 '*     FM-8 ﾊｲﾚﾂ(ｾｲｽｳｶﾞﾀ)ﾃﾞｰﾀ <-> ﾏｼﾝｺﾞ ﾃﾞｰﾀ ﾍﾝｶﾝ ｻﾌﾞﾙｰﾁﾝ                        *
120 '**************************************************************************
130 '
140 DIM A%(100) '(1)   (2) ﾉ ﾊﾞｱｲ ﾊｲﾚﾂｻｲｽﾞｰ>ﾏｼﾝｺﾞﾃﾞｰﾀ ｻｲｽﾞ
150 '*** (1) ﾊｲﾚﾂ -> ﾏｼﾝｺﾞﾃﾞｰﾀ     ******************************************
160 DEF USR0=&H7502
170 POKE &H7500,&H30:POKE &H7501,&H00'ﾏｼﾝｺﾞ ﾃﾞｰﾀ ｼﾖｳ ADDRESS
180 W=USR0(VARPTR(A%(0)))
190 '*** (2) ﾊｲﾚﾂ <- ﾏｼﾝｺﾞﾃﾞｰﾀ     ******************************************
200 DEF USR1=&H7517
210 POKE &H7500,&H30:POKE &H7501,&H00'ﾏｴ ﾄ ｵﾅｼﾞﾅﾗﾌﾖｳ
220 W=USR1(VARPTR(A%(0)))
230 '*** (3) ﾏｼﾝｺﾞﾃﾞｰﾀ ﾁｮｸｾﾂ PUT  (ﾄﾞｯﾄ ﾉﾐ) ***********************************
240 DEF USR2=&H7524
260 A%(0)=0 'X-START
261 A%(1)=50 'Y-START
262 A%(2)=40 'X-END
270 A%(3)=80 'Y-END
271 A%(4)=&H3000'START ADDRESS
272 A%(5)=&H3100'END   ADDRESS
280 A%(6)=0 '@A
281 A%(7)=0 'ｷﾉｳ
282 A%(8)=7 'COLOR
290 W=USR2(VARPTR(A%(0)))
300 '**************************************************************************
```

FM-8 カラーパターン・エディタ

ETF ラボラトリ■笠作貴弥

I/O誌'81年12月号の『MZ-80B用TV黒板プログラム』と'82年1月号表紙の『FM-8グラフィック花札』の絵のすばらしさに刺激されて，FM-8用パターン・エディタを作って見ました．

使い方

❶プログラムをRUNすると，画面をイニシャライズしたあと，パターンの大きさを聞いてくるので，タテ，ヨコの順で答えてください．最大タテ16×ヨコ64ドットです．それ以上の大きさのものを作りたい場合は，いくつかに分けて行なうとよいと思います．
❷あとは，画面にマトリックスが表示されるので，各コマンドを活用して，パターンを作成してください．
❸パターンが完成したら，Rコマンドでパターンのデータを BASICのデータ文として作成します．Rコマンドを指定すると，**"カラーモード？"**と聞いてくるので，カラーパラメータとともにデータにする場合には，**Y**を，そうでないときは，他のキーを押してください．

次に，**"ギョウバンゴウ？"**と聞いてくるので，データ文を作成する先頭の行番号を入力してください．ただし，1000～31999の間に限ります．後で，簡単にリナンバーができるので，その間ならいくつでもよいと思います．

次は，**"ヘンスウメイ？"**と聞いてくるので，後々このデータを読み込む配列の変数名を入力してください．
❹以上，入力が終わると，行番号を表示しながらデータ文を自動作成して，このプログラム自身にアペンドします．ベルがピッピッピッ‥‥と鳴ると作成完了で，再びプログラムがスタートします．

ストップ・キーを押して，リストを見てください．みごとにデータ文および，それを配列に読み込むサブルーチン・プログラムが，自動作成されていると思います．あとは，本プログラムをデリートして，メインのプログラムを作成してください．

コマンドの説明

[1]～[7]：ペン・カラーを設定します．

[←][↑][↓][→]：カーソルを移動します．ペン・ダウンの状態では，カーソルの移動とともにドットをセットまたは，リセットします．

[S]：現在のカーソル位置のドットをセットした後，カーソルをX方向に＋1進めます．

[SPACE]：現在のカーソル位置のドットをリセットした後，カーソルをX方向に＋1進めます．

[A]：すべてのドットを現在のペン・カラーでセットします．

[C]または[CLEAR]：すべてのドットをリセットします．

[U]：ペンをUPします．

[D]：ペンをDOWNします．

[M]：ペン・ダウン状態のとき，カーソル移動でドットをセットするか，リセットするか，モード変換します．

[R]：ドット・データをBASICのデータ文として，自動作成します．

プログラムについて

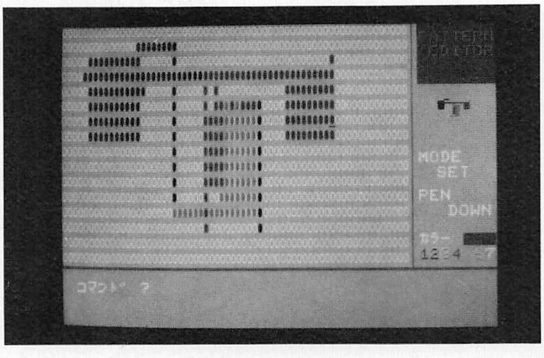
フローチャートも書かずにいきなりコーティングしたので，非常に解りにくいプログラムになってしまいました．

最も苦労したのは，データ文を自動作成して，アペンドする部分で，ここだけで5回以上コーティングをやり直しました．

具体的にどうやって，アペンドするかと言うと，最初にドット・パターンを配列に読み込んだ後，メモリの＄6100番地以降に実際のテキストが，メモリ上に格納されているのと同じように，データを書き込んでいきます．

そのあと，マシン語により実際のテキストの後へブロック転送して，各種ポインタを書き換えています．

大変わかりにくいと思いますが，ファイトのある人はプログラムを読んでみてください．



あとがき

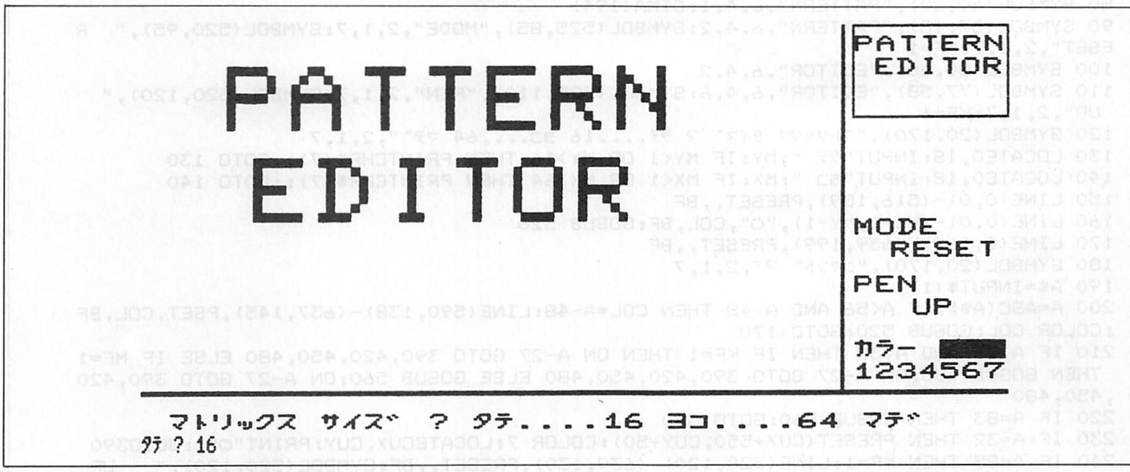
いきなりコーティングしたわりには，結構便利なプログラムができたと思っています．しかし，一度に作成できるパターンの大きさが小さいので，今後，改良していきたいと思います．なお，ディスクBASICでも動作確認しています．

◘参考文献

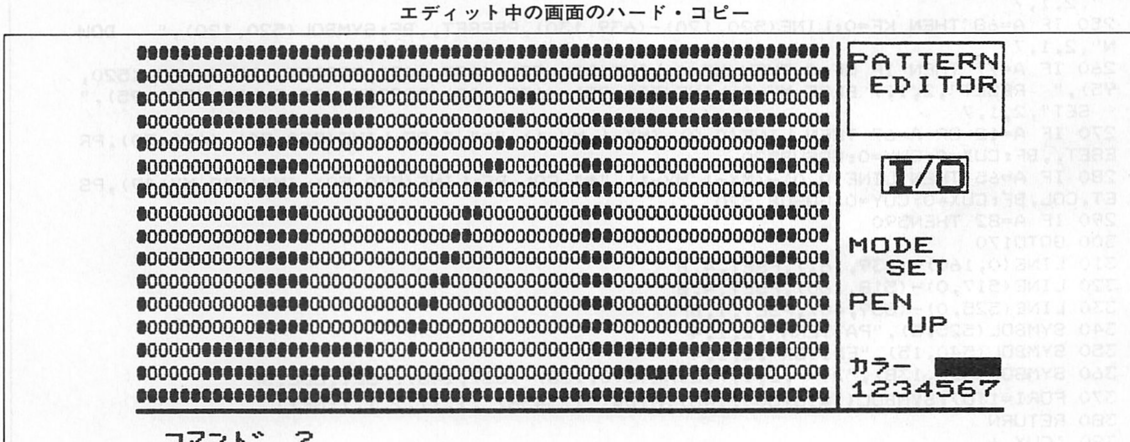
1）高嶋　晃：“TV黒板”，I/O，’81年12月号

2）F-BASIC文法書

初期画面のハード・コピー



エディット中の画面のハード・コピー



自動作成されるプログラム例

```
2000 FOR I=0 TO 194:READ A%(I):NEXT:RETURN
2010 DATA -1,-1,-1,-1,-32768,0,0,1,-24577,-1024,12351,-31
2020 DATA -32708,0,24688,113,-32708,0,-16272,113,-32708,1,-32656,113
2030 DATA -32708,3,112,113,-32708,6,112,113,-32708,12,112,113
2040 DATA -32708,24,112,113,-32708,48,112,113,-32708,96,112,113
2050 DATA -32708,192,112,113,-24577,-896,63,-31,-32768,0,0,1
2060 DATA -1,-1,-1,-1,0,0,0,0,0,0,0,0
2070 DATA 0,0,0,0,0,0,0,0,0,0,0,0
2080 DATA 0,0,0,0,0,0,0,0,0,0,0,0
2090 DATA 0,0,0,0,0,0,0,0,0,0,0,0
2100 DATA 0,0,0,0,0,0,0,0,0,0,0,0
2110 DATA 0,0,0,0,0,0,0,0,0,0,0,0
2120 DATA 0,0,0,0,0,0,0,0,0,0,0,0
2130 DATA 0,0,0,0,0,0,0,0,0,0,0,0
2140 DATA 0,0,0,0,0,0,0,0,0,0,0,0
2150 DATA 0,0,0,0,0,0,0,0,0,0,0,0
2160 DATA 0,0,0,0,0,0,0,0,0,0,0,0
2170 DATA 0,0,0
```

カラーパターン・エディタ プログラム・リスト

```
10 '*********************************************************************
20 '*                                                                   *
30 '*  PATTERN EDITOR    Copyright 1982 By ETF Laboratory T.KASASAKU *
40 '*                                                                   *
50 '*********************************************************************
60 WIDTH 80,20:CLEAR 300,&H5FFF:DEFINT A-Z:COL=7:COLOR COL
70 GOSUB 310
80 SYMBOL(60,20),"PATTERN",6,4,1:DIMA(194)
90 SYMBOL(57,18),"PATTERN",6,4,2:SYMBOL(525,85),"MODE",2,1,7:SYMBOL(520,95),"  R
ESET",2,1,7:MF=1
100 SYMBOL(80,60),"EDITOR",6,4,2
110 SYMBOL(77,58),"EDITOR",6,4,6:SYMBOL(525,110),"PEN",2,1,7:SYMBOL(520,120),"
 UP",2,1,7:KF=1
120 SYMBOL(20,170),"ﾏﾄﾘｯｸｽ ｻｲｽﾞ ? ﾀﾃ....16 ﾖｺ....64 ﾏﾃﾞ",2,1,7
130 LOCATE0,18:INPUT"ﾀﾃ ";MY:IF MY<1 OR MY>16 THEN PRINTCHR$(7);:GOTO 130
140 LOCATE0,18:INPUT"ﾖｺ ";MX:IF MX<1 OR MX>64 THEN PRINTCHR$(7);:GOTO 140
150 LINE(0,0)-(516,159),PRESET,,BF
160 LINE(0,0)-(MX-1,MY-1),"○",COL,BF:GOSUB 520
170 LINE(0,162)-(639,199),PRESET,,BF
180 SYMBOL(20,170),"ｺﾏﾝﾄﾞ ?",2,1,7
190 A$=INPUT$(1)
200 A=ASC(A$):IF A<56 AND A>48 THEN COL=A-48:LINE(590,138)-(637,145),PSET,COL,BF
:COLOR COL:GOSUB 520:GOTO 170
210 IF A>27 AND A<32 THEN IF KF=1 THEN ON A-27 GOTO 390,420,450,480 ELSE IF MF=1
 THEN GOSUB 580:ON A-27 GOTO 390,420,450,480 ELSE GOSUB 560:ON A-27 GOTO 390,420
,450,480
220 IF A=83 THEN GOSUB 560:GOTO 390
230 IF A=32 THEN PRESET(CUX+550,CUY+50):COLOR 7:LOCATECUX,CUY:PRINT"○";:GOTO390
240 IF A=85 THEN KF=1:LINE(520,120)-(639,130),PRESET,,BF:SYMBOL(520,120),"   UP
",2,1,7
250 IF A=68 THEN KF=0:LINE(520,120)-(639,130),PRESET,,BF:SYMBOL(520,120),"   DOW
N",2,1,7
260 IF A=77 THEN IF MF=0 THEN MF=1:LINE(520,95)-(639,105),PRESET,,BF:SYMBOL(520,
95),"  RESET",2,1,7 ELSE MF=0:LINE(520,95)-(639,105),PRESET,,BF:SYMBOL(520,95),"
  SET",2,1,7
270 IF A=12 OR A=67 THEN LINE(0,0)-(MX-1,MY-1),"○",7,BF:LINE(550,50)-(620,70),PR
ESET,,BF:CUX=0:CUY=0:GOSUB520
280 IF A=65 THEN LINE(0,0)-(MX-1,MY-1),"●",COL,BF:LINE(550,50)-(MX+549,MY+49),PS
ET,COL,BF:CUX=0:CUY=0:GOSUB 520
290 IF A=82 THEN590
300 GOTO170
310 LINE(0,160)-(639,161),PSET,4,B
320 LINE(517,0)-(518,160),PSET,4,B
330 LINE(525,0)-(639,40),PSET,1,BF
340 SYMBOL(525,5),"PATTERN",2,1,4
350 SYMBOL(540,15),"EDITOR",2,1,4
360 SYMBOL(525,138),"ｶﾗｰ",2,1,7:LINE(590,138)-(637,145),PSET,COL,BF
370 FORI=1TO7:SYMBOL(I*16+509,148),MID$("1234567",I,1),2,1,I:NEXT
380 RETURN
390 'CUX +
400 GOSUB540:CUX=CUX+1:IF CUX=MX THEN CUX=0:CUY=CUY+1:IF CUY=MY THEN CUY=0
410 GOSUB520:GOTO190
420 'CUX -
430 GOSUB540:CUX=CUX-1:IF CUX<0 THEN CUX=MX-1:CUY=CUY-1:IF CUY<0 THEN CUY=MY-1
440 GOSUB520:GOTO190
450 'CUY +
460 GOSUB540:CUY=CUY-1:IF CUY<0 THEN CUY=MY-1
470 GOSUB520:GOTO190
480 'CUY -
490 GOSUB540:CUY=CUY+1:IF CUY=MY THEN CUY=0
500 GOSUB520:GOTO190
510 'CURSOR SET
520 LINE(CUX*8,CUY*10+8)-(CUX*8+8,CUY*10+8),PSET,COL:RETURN
530 'CURSOR RESET
540 LINE(CUX*8,CUY*10+8)-(CUX*8+8,CUY*10+8),PRESET:RETURN
550 'DOT SET
560 PSET(CUX+550,CUY+50,COL):COLORCOL:LOCATECUX,CUY:PRINT"●";:RETURN
570 'DOT RESET
580 PRESET(CUX+550,CUY+50):COLOR7:LOCATECUX,CUY:PRINT"○";:RETURN
590 LINE(0,162)-(639,199),PRESET,,BF
600 SYMBOL(20,170),"ｶﾗｰ ﾓｰﾄﾞ ? (Y/N)",2,1,7:A$=INPUT$(1)
610 IF A$="Y" THEN C=(((MX*MY+7)/8)*3+2-1)/2:GET@A(550,50)-(MX+549,MY+49),A,G EL
SE C=((MX*MY+7)/8+2-1)/2:GET@(550,50)-(MX+549,MY+49),A,G
620 LOCATE0,18:INPUT "ｷﾞｮｳ ﾊﾞﾝｺﾞｳ ";GYO:IF GYO<1000 OR GYO>32000 THEN PRINTCHR$(
```

```
7);:GOTO620
630 LOCATE0,18:INPUT "ﾍﾝｽｳ ﾒｲ       ";H$:IF ASC(H$)<65 OR ASC(H$)>90 THEN PRINTCHR
$(7);:GOTO630
640 LINE(0,161)-(639,199),PRESET,,BF:LOCATE0,17
650 'MACHINE LANGUAGE
660 FORI=&H6000 TO &H6023:READ A$:POKE I,VAL("&H"+A$):NEXT
670 DATA 34,72,10,BE,60,50,BE,00,35,30,1E,CE,61,00,A6,C0,A7,B0
680 DATA 31,3F,26,F8,BE,60,52,BF,00,35,BF,00,3B,BF,00,3D,35,F2
690 'FOR DATA
700 DATA 81,20,49,E6,FE,01,00,20,CD,20,FE,01,95,3A,85,20,25
710 DATA 28,49,29,3A,82,3A,8B,0
720 LP=PEEK(&H35)*256+PEEK(&H36)-2:PO=&H6100:LP1=PO
730 GOSUB930:PO=PO+1:LP=LP+1
740 FORI=1TO16:READ A$:POKE PO+I,VAL("&H"+A$):NEXT:PO=PO+I:LP=LP+I
750 FORI=1TO LEN(H$):POKE PO+I-1,ASC(MID$(H$,I,1)):NEXT:PO=PO+I-1:LP=LP+I-1
760 FORI=1TO9:READ A$:POKE PO+I-1,VAL("&H"+A$):NEXT:PO=PO+I-1:LP=LP+I-1
770 POKE &H6110,C:GOSUB960:I=0
780 GOSUB930
790 PO=PO+2:LP=LP+2:POKE PO,131:POKE PO+1,32:PO=PO+2:LP=LP+2
800 X=1
810 WHILE X<13 AND I<=C
820 A$=STR$(A(I)):IF A(I)>=0 THEN A$=RIGHT$(A$,LEN(A$)-1)
830 FORJ=1TO LEN(A$):POKE PO+J-1,ASC(MID$(A$,J,1)):NEXT:PO=PO+J-1:LP=LP+J-1
840 IF X<>12 AND I<>C THEN POKE PO,44:PO=PO+1:LP=LP+1
850 I=I+1:X=X+1
860 WEND
870 POKE PO,0:PO=PO+1:LP=LP+1:GOSUB960
880 IF I<=C THEN 780
890 POKE PO,0:POKE PO+1,0:PO=PO+2:LP=LP+2
900 POKE &H6050,INT((PO-&H60FF)/256):POKE &H6051,(PO-&H60FF) MOD 256
910 POKE &H6052,INT(LP/256):POKE &H6053,LP MOD 256
920 EXEC&H6000:FORI=1TO20:BEEP1:FORJ=1TO30:NEXT:FORJ=1TO30:BEEP0:NEXTJ,I:RUN
930 'LINE No. SET
940 PO=PO+2:LP=LP+2:POKE PO,INT(GYO/256):POKE PO+1,GYO MOD 256
950 PRINT"[";GYO;"]";:GYO=GYO+10:RETURN
960 ' ﾘﾝｸ POINTER SET
970 POKE LP1,INT(LP/256):POKE LP1+1,LP MOD 256:LP1=PO:RETURN
980 '--- END --- END --- END --- END --- END --- END --- END --- END --- END -
```

FM-8

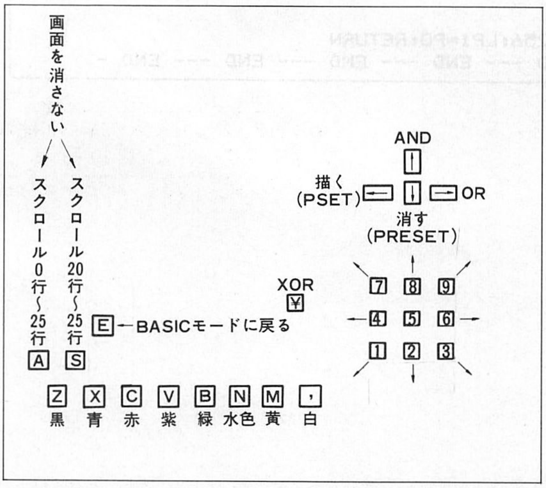
テレビ黒板

■滝沢吉人

このプログラムの特徴はキーボード操作によって任意の曲線を描くときにだけマシン語のサブルーチンを使います．マシン語サブルーチンでは点のX，Y座標を右下端に，常に表示します．また，キーボードの[S]，[A] を押すことで画面を消さないでスクロールを0行〜25行と20行〜25行自由に変えられます．マシン語モードのときのキーボードの操作は図1です．

通常はBASICコマンドが充実しているので，直接命令によって円などを描きます．

図1



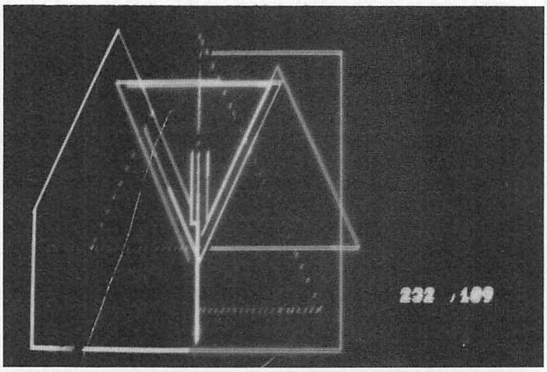


図2．BASICプログラムの説明

RUN	マシン語プログラムを読み込み，スクロールを20行から25行にします．
RUN 20	マシン語サブルーチンに飛び，点や線を描きます．
RUN 30	画面（0, 0)−(639, 161) をテープに記憶します．
RUN7070	テープに記憶されている画面データをディスプレイに描がきます．
RUN 200	画面（0, 0)−(639, 161) をテープに記憶します．
RUN 270	テープに記憶されている画面データをディスプレイ画面に表示します．

プログラムについて

BASICプログラムの説明は図2のとおりです．画面データ（39Kバイト）をカセットテープに記録するのに時間がかかるので，**PRINT＃1,B$** で1回に124バイト送る方法（**PRINT＃1, A%(N)** では2バイト）と，**GET@A** をしたあと，A%（0）のアドレスを **VARPTR(A%(0))** により調べ，**LOADM** によっていっぺんにテープにＳＡＶＥする方法を考えましたが，いずれも＆Ｈ8000以上のデータを配列Ａ%（Ｎ）に読み込ませることができなくて壁にぶつかってしまいました．

画面の転送について

■方式1

まず画面を **GET@A** により **DIMA%** に読み込み，**VARPTR**により配列Ａ%のアドレスを調べ，カセットテープに配列Ａ%のアドレスと配列Ａ%のデータを **SAVEM** によりテープにセーブします．

カセットテープからのロードはその手順を逆に行ないます．

■方式2

まず画面データを **GET@A** により **DIMA%(6482)** に読み込みます．この配列Ａ%のうち0でないものだけを16進

変換し，それをB＄にまとめて **PRINT＃1, B$** により一遍に 258 バイトずつカセットテープに転送します．

カセットテープからのＦＭ-８へのロードは上記の手順を逆に行います．

画面のテープへの SAVE は **GET＠A** で配列Ａ％(6482)に読み込んだデータを **PRINT＃1, A% (N)** で送ると50分ぐらいかかりますが，**方式１**ですと７分，**方式２**ですと６分に短縮されます．

テレビ黒板　BASICプログラム・リスト

```
10 WIDTH40,25:CONSOLE20,5,0,1:CLEAR,&H7BFA:LOADM"T":END
20 EXEC&H7BFB:END
30 CLEAR,&H7BFA:DIMA%(6482)
40 FOR Y=0TO161STEP54:GET@A(0,Y)-(639,Y+53),A%,G
50 OPEN"O",1,"F": S=VARPTR(A%(0)):PRINT#1,S:CLOSE
60 SAVEM"DATA",S,VARPTR(A%(6482))+1,S:NEXTY:END
70 CLEAR,&H7BFA:DIMA%(6482):FOR P=0 TO6482:A%(P)=1:NEXTP
80 FORY=0TO161STEP54: OPEN"I",1,"F":INPUT#1,S:CLOSE
90 T=VARPTR(A%(0))
100 LOADM"DATA",T-S
110 PUT@A(0,Y)-(639,Y+53),A%,PSET:NEXTY:END
150 PUT@A(0,Y)-(639,Y+31),A%,PSET:NEXTY:END
200 CLEAR1000,&H7BFA:OPEN"O",1,"F":DIMA%(6482)
210 FOR Y=0 TO161STEP54:GET@A(0,Y)-(639,Y+53),A%,G:FORN=0TO6482
220 IFA%(N)=0GOTO250
221 N$=RIGHT$("0"+"0"+"0"+HEX$(N),4):A$=RIGHT$("0"+"0"+"0"+HEX$(A%(N)),4)
230 B$=B$+N$+A$:K=K+1:IFK<31GOTO250
240 B$=B$+"TAKI":PRINT#1,B$:B$="":K=0
250 NEXTN:B$=B$+"ZAWA":PRINT#1,B$:B$="":K=0
260 NEXT Y:END
270 CLEAR1000,&H7BFA:DIMA%(6482):OPEN"I",1,"F"
280 FORY=0TO161STEP54
290 B$="":INPUT#1,B$:N=0
300 C$=MID$(B$,8*N+1,4):IFC$="TAKI"GOTO290
310 IFC$="ZAWA"GOTO350ELSED%=VAL("&H"+MID$(B$,8*N+5,2))
320 IF D%<128GOTO330ELSED%=D%-256
330 A%(VAL("&H"+C$))=256*D%+VAL("&H"+MID$(B$,8*N+7,2)):N=N+1
340 GOTO300
350 PUT@A(0,Y)-(639,Y+53),A%,PSET
360 IFY>100GOTO380
370 FORZ=0TO6482:A%(Z)=0:NEXTZ
380 NEXTY:END
```

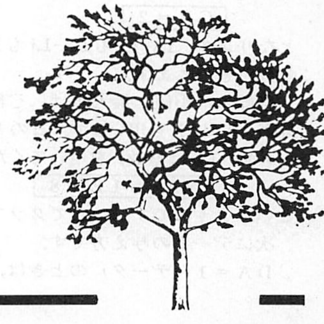
テレビ黒板　マシン語・ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7BF0 00 00 00 00 00 00 00 00 00 00 00 86 00 B7 7D CA :84
---------------------------------------------------------
Sum: 00 00 00 00 00 00 00 00 00 00 00 86 00 B7 7D CA :84

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7C00 34 3F BD DD 69 81 34 26 0B BE 7C EF 30 1F BF 7C :0F
7C10 EF 7E 7C 8B 81 36 26 0B BE 7C EF 30 01 BF 7C EF :E0
7C20 7E 7C 8B 81 38 26 06 7A 7C F2 7E 7C 8B 81 38 26 :B6
7C30 06 7A 7C F2 7E 7C 8B 81 32 26 06 7C 7C F2 7E 7C :36
7C40 8B 81 37 26 0E 7A 7C F2 BE 7C EF 30 1F BF 7C EF :01
7C50 7E 7C 8B 81 39 26 0E 7A 7C F2 BE 7C EF 30 01 BF :74
7C60 7C EF 7E 7C 8B 81 31 26 0D 7C 7C F2 BE 7C EF 30 :18
7C70 1F BF 7C EF 20 15 81 33 26 0E 7C 7C F2 BE 7C EF :79
7C80 30 01 BF 7C EF 7E 7C 8B BD 7D 1D FC 7C EF BD 9B :F6
7C90 EF BD C0 83 BD B9 78 EC 01 FD 7D 04 EC 03 FD 7D :B1
7CA0 06 FC 7C F1 BD 9B EF BD C0 83 BD B9 78 EC 01 FD :8E
7CB0 7D 09 A6 03 B7 7D 0B 34 08 86 FD 1F 8B 8E 7C F5 :D6
7CC0 BD F6 25 F6 7C F4 34 04 8E 7C E5 57 24 06 7F 7C :E1
7CD0 F4 BD F6 25 35 04 F7 7C F4 BD F6 25 35 08 7E 7D :7C
7CE0 BF 00 00 00 00 10 00 7C EB 00 0A 00 00 17 01 01 :59
```

```
7CF0 41 00 2F 04 04 10 00 7C FB 00 11 00 00 07 20 14 :4B
---------------------------------------------------------
Sum: 9E D4 E7 FF 67 F6 40 D1 D2 06 DE 85 BA 12 2E F2 :ED

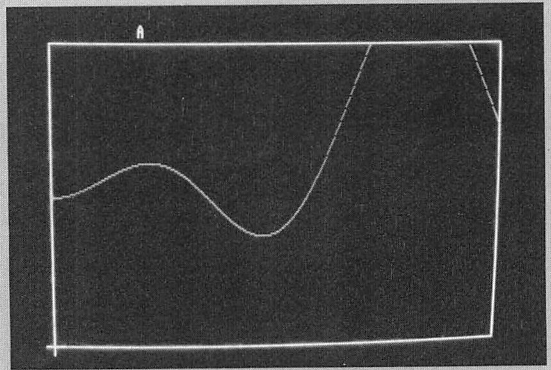
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7D00 27 14 07 20 33 32 31 00 2C 34 37 00 10 00 7D 12 :2E
7D10 00 0B 00 00 01 00 28 19 14 05 00 00 00 81 1D 26 :2A
7D20 04 7F 7C F4 39 81 1F 26 06 86 01 B7 7C F4 39 81 :60
7D30 1C 26 06 86 02 B7 7C F4 39 81 1E 26 06 86 03 B7 :3B
7D40 7C F4 39 81 5C 26 06 86 04 B7 7C F4 39 81 5A 26 :9D
7D50 04 7F 7C F3 39 81 58 26 06 86 01 B7 7C F3 39 81 :97
7D60 43 26 06 86 02 B7 7C F3 39 81 56 26 06 86 03 B7 :99
7D70 7C F3 39 81 42 26 06 86 04 B7 7C F3 39 81 4E 26 :75
7D80 06 86 05 B7 FC F3 39 81 4D 26 06 86 06 B7 7C F3 :1C
7D90 39 81 2C 26 06 86 07 B7 7C F3 39 81 41 26 0D 8E :81
7DA0 00 19 BF 7D 18 8E 7D 0C BD F2 D8 39 81 53 26 05 :43
7DB0 8E 14 05 20 ED 81 45 26 05 86 FF B7 7D CA 39 7D :DE
7DC0 7D CA 26 03 7E 7C 02 35 3F 39 FF 00 00 00 00 00 :18
---------------------------------------------------------
Sum: D0 4E 98 92 CD F2 D8 F7 90 7F BA 9B CB 70 A2 F7 :0B
```



FM-8

グラフ作成プログラム

■児玉良幸



グラフ作成というのは，なかなか面倒な作業です．データの数が多ければなおのこと，また，ある数式がどんなグラフになるのか直感するのは，困難なことです．そこでグラフ作成プログラムを作りました．

1 応用範囲は広い方がいい

このプログラムの特徴は，あらゆるプログラムのサブルーチンとして，自分のほしい型のグラフが作成できるという点にあります．

選択型式は，表1のようになります．

表1　グラフ選択

- データ様式 ― データ / 数式
- グラフの型 ― Linear-Linear / Linear-Log / Log-Linear / Log-Log

2 使い方がわからなくては話は始まらない

引数について説明します(表2参照)．

データの様式が2通りあるということを，先に述べましたが，その様式の与え方は，変数DAを使います．たとえば，数式で与える場合は，

DA＝2

とします．

グラフの型は，変数GRです．X座標をLinear，Y座標をLogと指定するときは，

GR＝2

となります．Li-LiもLo-LiもLo-Loも同様に，それぞれの数値を与えます．

座標の最小値，最大値は，それぞれXMIN，XMAX，YMIN，YMAXを用います．このときLOG座標に関しては，1×10^n（nは整数）を用いてください．たとえば

YMAX＝1E－8

というようにです．これでグラフの型は決まりました．

次にデータの与え方です．

DA＝1（データ）のときは，データの数をNDに代入

表2　引数および数式

DA	データの様式	1 データ 2 式
GR	グラフの型	1 Linear-Linear 2 Linear-Log 3 Log-Linear 4 Log-Log
XD（　）	DA＝1のときのXデータ	
YD（　）	DA＝1のときのYデータ	
ND	DA＝1のときのデータ数	
XMIN	X座標最小値	
XMAX	X座標最大値	
YMIN	Y座標最小値	
TMAX	Y座標最大値	
DEF FNY(X)	＝DA＝2のときの式を与える．	

し，Xデータ，Yデータをそれぞれ，

XD（　）＝
YD（　）＝

というように，ディメンジョンで与えます．当然，最初にDIM YD(ND), XD(ND)を宣言する必要があります．

DA＝2（数式）のときは，式を

DEF FNY(X)　＝

で与えてください．

準備ができたら，GOSUB 10000で，自動的に枠を描き，目盛を打ちます．

3 プログラムを見てみようか

リスト1が，グラフ作成サブルーチン・プログラムです．

10000～10140	SUBROUTNEの初期設定です．
10150～10200	X座標の目盛を計算により適当に作り出します．
10210～10280	Y座標の目盛を計算により適当に作り出します．
10400～10520	10150～10280で決めた目盛のデータに従ってグラフの枠を作成します．
15000～15090	データ形式のときの点のプロット・サブルーチン
16000～16110	式のときの点のプロット・サブルーチン

20000〜20140	linear scaleを作り出すサブルーチン
22000〜22130	log scaleを作り出すサブルーチン

4 サブルーチンがあってもメインがなければ

簡単な応用例を示します．

❶数式をグラフにする（**リスト2**）．

最も基本的な使い方として数式をグラフにしてみましょう．必要なデータを与え終わると，画面にグラフを作成します．

出力結果1〜3は，**DEF FNY(X)＝X'^2＋Sin(X)＊10＊X**によって，Li-Li，Li-Lo，Lo-Loのグラフを作ったものです．

行番号10を書き変えることにより，いろいろな数式を与えることができます．グラフができたところで，[h]を入力すると，**Hardc 1**が実行され[H]を入力すると，**Hardc2**が，[E]の入力によりプログラムが終わります．

❷データをキーボードより入力する（**リスト3**）．

説明は省きます．指示に従って，入力をすればグラフができます．

5 終わりは悲しくやってくる

このプログラムを利用して，私は金属の内部摩擦のデータ整理をしています(出力結果4)．データから各種の計算を行なわせ，データを外部記憶装置に送り，最後にこのプログラムを利用して，様々なデータの組み合わせを呼び出グラフにしています．

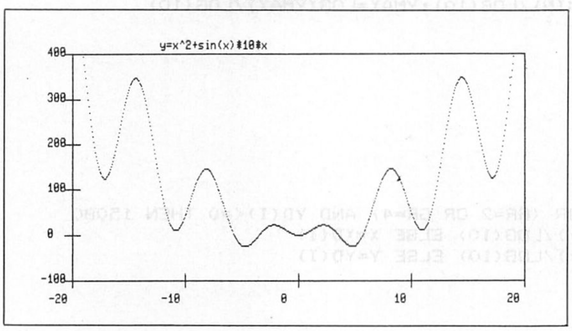
このプログラムは，少しの変更により，極座標にすることも可能です．

メイン・プログラムとしては，計測装置からのデータ読み込みや，FM-8の持っている4chアナログ入力を使用するのも面白いでしょう．

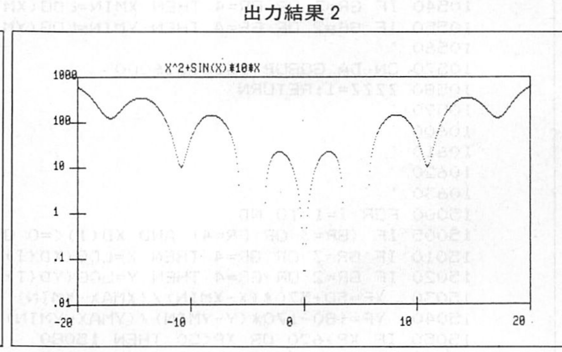
最後に何かと助言をいただいた，研究室の西山先生にお礼を申しあげます．

◘参考文献
F-BASIC文法書

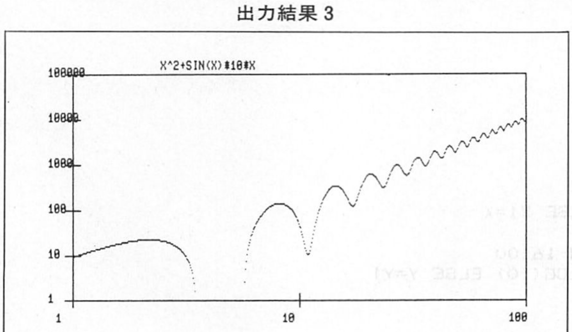
出力結果1



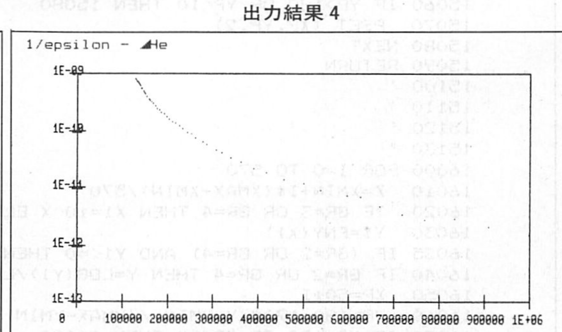
出力結果2



出力結果3



出力結果4



リスト1

```
10000 '*****************************************************
10010 '*                                                   *
10020 '*    subroutine       GRAPH        Ver 1.0          *
10030 '*                                                   *
10040 '*                   (by  Y.Kodama)                  *
10050 '*                                                   *
10060 '*****************************************************
10090 '
10100 'main
10110 IF ZZZZ=1 THEN 10130
10120 DIM XL(20),XL$(20),YL(20),YL$(20),L(20),L$(20)
10130 WIDTH 80,25
```

リスト1

```
10140 CONSOLE 0,25
10150 MAX=XMAX:MIN=XMIN
10160 ON GR GOSUB 20000,20000,22000,22000
10170 FOR I=0 TO C-1
10180 XL(I)=L(I)
10190 XL$(I)=L$(I)
10200 NEXT
10210 XN=C-1
10220 MAX=YMAX:MIN=YMIN
10230 ON GR GOSUB 20000,22000,20000,22000
10240 FOR I=0 TO C-1
10250 YL(I)=L(I)
10260 YL$(I)=L$(I)
10270 NEXT
10280 YN=C-1
10400 '
10410 'make graph
10420 '
10430 LINE (50,10)-(620,180),PSET,4,B
10440 FOR I=0 TO XN
10450 LINE (50+XL(I)*570,175)-(50+XL(I)*570,185),PSET,4
10460 SYMBOL (20+XL(I)*570,190),XL$(I),1,1,7
10470 NEXT
10480 '
10490 FOR I=0 TO YN
10500 LINE (40,180-YL(I)*170)-(60,180-YL(I)*170),PSET,4
10510 SYMBOL (10,175-YL(I)*170),YL$(I),1,1,7
10520 NEXT
10530 '
10540 IF GR=3 OR GR=4 THEN XMIN=LOG(XMIN)/LOG(10):XMAX=LOG(XMAX)/LOG(10)
10550 IF GR=2 OR GR=4 THEN YMIN=LOG(YMIN)/LOG(10):YMAX=LOG(YMAX)/LOG(10)
10560 '
10570 ON DA GOSUB 15000,16000
10580 ZZZZ=1:RETURN
10590 '
10600 '
10610 '
10620 '
10630 '
15000 FOR I=1 TO ND
15005 IF (GR=3 OR GR=4) AND XD(I)<=0 OR (GR=2 OR GR=4) AND YD(I)<=0 THEN 15080
15010 IF GR=3 OR GR=4 THEN X=LOG(XD(I))/LOG(10) ELSE X=XD(I)
15020 IF GR=2 OR GR=4 THEN Y=LOG(YD(I))/LOG(10) ELSE Y=YD(I)
15030  XP=50+570*(X-XMIN)/(XMAX-XMIN)
15040  YP=180-170*(Y-YMIN)/(YMAX-YMIN)
15050 IF XP>620 OR XP<50 THEN 15080
15060 IF YP>180 OR YP<10 THEN 15080
15070  PSET (XP,YP,2)
15080 NEXT
15090 RETURN
15100 '
15110 '
15120 '
15130 '
16000 FOR I=0 TO 570
16010  X=XMIN+I*(XMAX-XMIN)/570
16020  IF GR=3 OR GR=4 THEN X1=10^X ELSE X1=X
16030  Y1=FNY(X1)
16035 IF (GR=2 OR GR=4) AND Y1<=0 THEN 16100
16040 IF GR=2 OR GR=4 THEN Y=LOG(Y1)/LOG(10) ELSE Y=Y1
16050  XP=50+I
16060  YP=180-170*(Y-YMIN)/(YMAX-YMIN)
16070 IF XP>620 OR XP<50 THEN 16100
16080 IF YP>180 OR YP<10 THEN 16100
16090  PSET (XP,YP,2)
16100 NEXT
16110 RETURN
16120 '
16130 '
16140 '
16150 '
20000 'subroutine linear scale
20010 '
20020 L(0)=0
20030 L(1)=1
20040 L$(0)=STR$(MIN)
20050 L$(1)=STR$(MAX)
20060 K1=INT(LOG(MAX-MIN)/LOG(10)-.1)
20070 M1=INT((MIN/10^K1)+1)*10^K1
```

リスト1

```
20080 C=2
20090 FOR ME=M1 TO MAX STEP 10^K1
20095 ME=INT(ME/10^K1+.5)*10^K1
20100  L(C)=(ME-MIN)/(MAX-MIN)
20110  L$(C)=STR$(ME)
20120   C=C+1
20130 NEXT
20140 RETURN
20150 '
20160 '
20170 '
22000 'subloutine log scale
22010 '
22020 L(0)=0
22030 L(1)=1
22040 L$(0)=STR$(MIN)
22050 L$(1)=STR$(MAX)
22060 K1=LOG(MAX/MIN)/LOG(10)
22070 C=2
22080 FOR ME=1/K1 TO 1 STEP 1/K1
22090 L(C)=ME
22100 L$(C)=STR$(MIN*10^(C-1))
22110  C=C+1
22120 NEXT
22130 RETURN
22140 '*********************************************
22150 '*                                           *
22160 '*        END        1981年 9月 17日 15時 56分 56秒   *
22170 '*                                           *
22180 '*********************************************
```

リスト2

```
1 '
2 'PROGRAM    No 1
3 '
4 '
5 WIDTH 80,20
6 CONSOLE 0,20,0,1
7 COLOR 4,0
8 CLS
10 DEF FNY(X)=
20 DA=2
30 LINE INPUT"ｸﾞﾗﾌ ﾉ ﾅﾏｴ ",A$
40 INPUT"xmin = ",XMIN
50 INPUT"xmax = ",XMAX
60 INPUT"ymin = ",YMIN
70 INPUT"ymax = ",YMAX
80 INPUT" gr  = ",GR
90 GOSUB 10000
100 LOCATE 20,0
110 PRINT A$
120 IF INKEY$="e" THEN CLS:END
130 IF INKEY$="h" THEN HARDC 1
140 IF INKEY$="H" THEN HARDC 2
150 GOTO 120
```

リスト3

```
10 WIDTH 80,25
20 CONSOLE 0,25,0,1
30 CLS
40 INPUT"ﾀﾞｲﾀｲ ﾉ ﾃﾞｰﾀｽｳ ";N
50 DIM XD(N+10),YD(N+10)
60 I=1
70 PRINT "x(";I;")=";:INPUT"",XD(I)
80 PRINT "y(";I;")=";:INPUT"",YD(I)
90 IF XD(I)=0 THEN XD(I)=XD(I-1)
100 IF YD(I)=0 THEN YD(I)=YD(I-1)
110 IF XD(I)=-999 THEN 130
120 I=I+1
125 GOTO 70
130 INPUT"1-ﾃﾞｰﾀ ｦ ﾅｵｽ  2-ﾆｭｳﾘｮｸ ｵﾜﾘ ";A
140 IF A=2 THEN 200
150 PRINT
160 INPUT"  I=";I
170 PRINT
180 GOTO 70
200 ND=I
210 INPUT"Xmin =",XMIN
220 INPUT"Xmax =",XMAX
230 INPUT"Ymin =",YMIN
240 INPUT"Ymax =",YMAX
250 INPUT "gr=",GR
260 DA=1
270 GOSUB 10000
280 IF INKEY$="h" THEN HARDC1
290 IF INKEY$="H" THEN HARDC2
300 IF INKEY$="e" THEN 210
310 GOTO 280
```

FM-8

●グラフの作成にべんりな…

Auto Plot

■岡田 皇三

Auto Plotはグラフ作成用プログラムです．画面をグラフ表示領域と，グラフの仕様を入力する領域とに分かれているので，グラフの仕様，データの変更が容易です．また，１つのグラフに数種類のデータを表示して比較することが可能です．非常に使いやすくなっているので，どんどん使ってみてください．

プログラムの説明

本プログラムのフローチャートを図１に示します．

各ブロックの内容を説明します．

SET

グラフの仕様を決めます．カーソル移動キーで編集します．

Ⓐ PLOT SPEC.

1．LINE(1〜3)：プロットするデータの線の種類．
　1…実線，２，３…間隔の異なる破線

2．MIN X：X軸の最小値

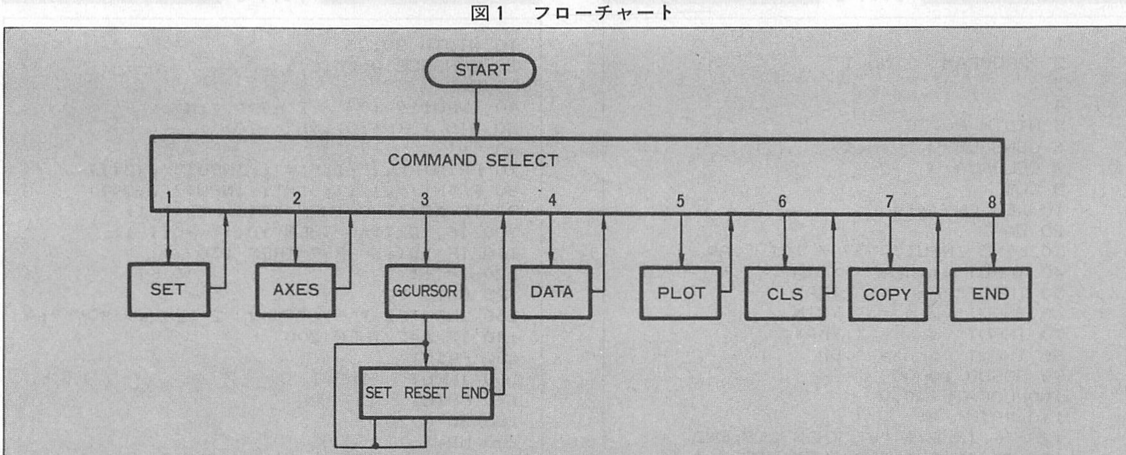
3．MAX X：X軸の最大値

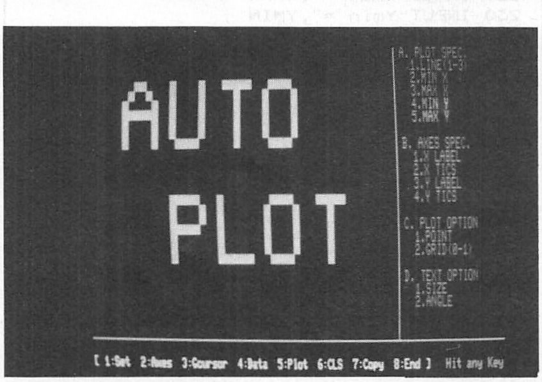
4．MIN Y：Y軸の最小値

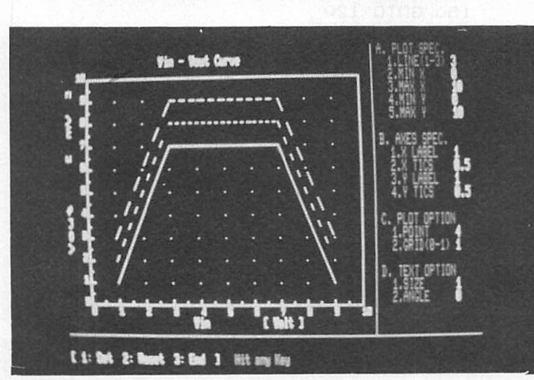
5．MAX Y：Y軸の最大値

Ⓑ. AXES SPEC.

図１　フローチャート







1．X LABEL：X軸の目盛の数値を表示する間隔．
2．X TICS ：X LABELを分割する最小単位の長さ．
3．Y LABEL ：Xのそれと同じ．
4．Y TICS ：Xのそれと同じ．

Ⓒ．PLOT OPTION

1．POINT：プロットするデータの数．
2．GRID（0－1）：1でX，YのLABEL の交点を表示．

Ⓓ．TEXT OPTION

1．SIZE：GCURSORで書き込む文字の大きさ，8ドット単位．
2．ANGLE：文字を回転する角度を示す．入力する数値は0～3です．カーソル移動KEYで編集します．

AXES

SETの仕様にしたがい，グラフのわくを表示します．

GCURSOR

GRAPHIC CURSOR を動かして文字を書き込んだり(SET)，消去したり(RESET)するのに使います．

DATA

PLOTするデータのX，Y値を入力します．“－”，“．”，“0”～“9”が有効で，最大5文字までです．なお，入力は左づめで入力してください．カーソル移動キーで編集します．↑：前のデータ表示，↓：次のデータ表示．

PLOT

DATAにしたがい，入力したデータをPLOTします．

CLS

480×180のグラフ表示領域を消去します．

COPY

480×180のグラフ表示領域のハード・コピーをプリンタに出力します．

END

プログラムの終了．

あとがき

このプログラムの使い方は，とくに難かしくないと思います．プログラムをRUNさせれば自然に解ります．このプログラムはBASICで書いていますが，COPYをのぞいてそれほど実効速度は気にならないと思います．しかし，COPYの実効時間約15分と長いので，今後機械語に直して実効速度をあげる必要があると思います．なお，私のプリンタはエプソンのMP-80 Kなので，他のプリンタを使っておられるかたはプログラムを少し変更する必要があるかもしれません．

Auto Plot BASICプログラム・リスト

```
1000 '********************************************************************
1002 '*****                                                        *****
1004 '*****                    AUTO PLOT                           *****
1006 '*****                                    by Kozo  OKADA      *****
1008 '*****                                                        *****
1010 '********************************************************************
1012 CLEAR 2000
1014 WIDTH 80,25:X=75:Y=1:XD=14:N=1:GXO=230:GYO=100:CONSOLE24,1
1016 POKE &H1E3,0:POKE &HFD00,&H40
1018 OPEN "O",#2,"LPTO:"
1020 DIM PAR$(7,19),SUM$(19),DT$(50,10),DX$(50),DY$(50),DX(50),DY(50),G%(2)
1022 DEF FNX(X)=40+((XG-40)¥NX)*X:DEF FNY(Y)=160-((160-YG)¥NY)*Y
1024 DEF FNPX(X)=40+INT((XG-40)/(MAX-MIX)*(DX(X)-MIX))
1026 DEF FNPY(X)=160-INT((160-YG)/(MAY-MIY)*(DY(X)-MIY))
1028 DEF FNM(X)=(PY(X+1)-PY(X))/(PX(X+1)-PX(X))
1030 DEF FNDPY(K,I)=PY(I)+INT((K-PX(I))*(PY(I+1)-PY(I))/(PX(I+1)-PX(I)))
1032 DEF FNDPX(K,I)=PX(I)+INT((K-PY(I))*(PX(I+1)-PX(I))/(PY(I+1)-PY(I)))
1034 DEF FNPX1(X)=40+INT((XG-40)/(MAX-MIX)*(X-MIX))
1036 DEF FNPY1(X)=160-INT((160-YG)/(MAY-MIY)*(X-MIY))
1038 A2%(0)=-925:A2%(1)=8064
1040 A3%(0)=25864:A3%(1)=20352
1042 A4%(0)=-1971:A4%(1)=-28800
1044 A5%(0)=-1922:A5%(1)=8064
1046 A6%(0)=6483:A6%(1)=-3968
1048 A7%(0)=-962:A7%(1)=8064
1050 A8%(0)=-961:A8%(1)=8064
1052 A9%(0)=-956:A9%(1)=17408
1054 A10%(0)=-897:A10%(1)=8064
1056 A11%(0)=-898:A11%(1)=8064
1058 A1%(0)=0:A1%(1)=512:                  ' [.]
1060 A0%(0)=62:A0%(1)=0:                   ' [-]
1062 LINE@(480,0)-(480,180),PSET,3,B
1064 LINE@(0,180)-(639,180),PSET,3,B:COLOR 4
1066 LOCATE61,0:PRINT"A. PLOT SPEC."
1068 LOCATE63,1:PRINT"1.LINE(1-3)"
1070 LOCATE63,2:PRINT"2.MIN X"
1072 LOCATE63,3:PRINT"3.MAX X"
1074 LOCATE63,4:PRINT"4.MIN Y"
1076 LOCATE63,5:PRINT"5.MAX Y"
1078 LOCATE61,7:PRINT"B. AXES SPEC."
1080 LOCATE63,8:PRINT"1.X LABEL"
1082 LOCATE63,9:PRINT"2.X TICS "
1084 LOCATE63,10:PRINT"3.Y LABEL"
1086 LOCATE63,11:PRINT"4.Y TICS"
1088 LOCATE61,13:PRINT"C. PLOT OPTION"
```

Auto Plot BASIC プログラム・リスト

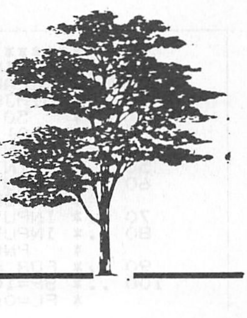
```
1090 LOCATE63,14:PRINT"1.POINT"
1092 LOCATE63,15:PRINT"2.GRID(0-1) "
1094 LOCATE61,17:PRINT"D. TEXT OPTION"
1096 LOCATE63,18:PRINT"1.SIZE "
1098 LOCATE63,19:PRINT"2.ANGLE"
1100 SYMBOL(50,20),"AUTO",9,6,6,0,PSET
1102 SYMBOL(122,90),"PLOT",9,6,6,0,PSET:COLOR 7
1104 LOCATE0,24:PRINT"[ 1:Set  2:Axes  3:Gcursor  4:Data  5:Plot  6:CLS  7:Copy
 8:End ] ";
1106 '
1108 '          COMMAND
1110 '
1112 SYMBOL(550,192),"Hit any Key",1,1,5
1114 CM$=INPUT$(1):IF CM$<"1" OR CM$>"8" THEN 1114
1116 CM=VAL(CM$):SYMBOL(550,192),"Hit any Key",1,1,0:ON CM GOSUB 1120,1228,1320,
1362,1464,1560,1530,1568
1118 GOTO 1104
1120 '
1122 '          SET
1124 '
1126 SYMBOL(504,168),"Hit E(End)",1,1,5
1128 LOCATE X,Y:IX=X-74:IY=Y:GOSUB 1156
1130 IN$=INPUT$(1):IN=ASC(IN$)
1132 IF (IN$<="9"ANDIN$>="0")OR (IN$="." OR IN$=CHR$(&H20)) OR IN$="-" THEN SYMB
OL(8*X,8*Y),CHR$(&H87),1,1,0:SYMBOL(8*X,8*Y),IN$,1,1:PAR$(IX,IY)=IN$:GOSUB 1170:
GOTO 1128
1134 IF IN>=28 AND IN<=31 THEN SYMBOL(8*X,8*Y),CHR$(&H87),1,1,,,XOR:ON IN-27 GOS
UB 1168,1184,1200,1216:GOTO 1128
1136 IF IN$="E" THEN SYMBOL(8*X,8*Y),CHR$(&H87),1,1,,,XOR:SYMBOL(504,168),"Hit E
(End)",1,1,0:GOSUB 1146:GOSUB 1148:RETURN
1138 GOSUB 1162:GOTO 1128
1140 '
1142 '          GET CHARACTER
1144 '
1146 SUM$(IY)="":FOR IX=1 TO 5:SUM$(IY)=SUM$(IY)+PAR$(IX,IY):NEXT:RETURN
1148 LT=VAL(SUM$(1)):MIX=VAL(SUM$(2)):MAX=VAL(SUM$(3)):MIY=VAL(SUM$(4))
1150 MAY=VAL(SUM$(5)):XL=VAL(SUM$(8)):XT=VAL(SUM$(9)):YL=VAL(SUM$(10))
1152 YT=VAL(SUM$(11)):PT=VAL(SUM$(14)):DG=VAL(SUM$(15)):SI=VAL(SUM$(18))
1154 AN=VAL(SUM$(19)):RETURN
1156 '
1158 '          PRINT
1160 '
1162 SYMBOL(8*X,8*Y),CHR$(&H87),1,1,,,XOR:RETURN
1164 '
1166 '          MOVE RIGHT
1168 '
1170 ON Y GOTO 1172,1172,1172,1172,1174,1178,1178,1172,1172,1172,1174,1178,1178,
1172,1174,1178,1178,1172,1176
1172 IF X=79 THEN GOSUB 1146:Y=Y+1:X=75:RETURN ELSE X=X+1:RETURN
1174 IF X=79 THEN GOSUB 1146:Y=Y+3:X=75:RETURN ELSE X=X+1:RETURN
1176 IF X=79 THEN GOSUB 1146:Y=1:X=75:RETURN ELSE X=X+1
1178 RETURN
1180 '
1182 '          MOVE LEFT
1184 '
1186 ON Y GOTO 1188,1190,1190,1190,1190,1194,1194,1192,1190,1190,1190,1194,1194,
1192,1190,1194,1194,1192,1190
1188 IF X=75 THEN GOSUB 1146:Y=19:X=79:RETURN ELSE X=X-1:RETURN
1190 IF X=75 THEN GOSUB 1146:Y=Y-1:X=79:RETURN ELSE X=X-1:RETURN
1192 IF X=75 THEN GOSUB 1146:Y=Y-3:X=79:RETURN ELSE X=X-1
1194 RETURN
1196 '
1198 '          MOVE UP
1200 '
1202 ON Y GOTO 1204,1206,1206,1206,1206,1210,1210,1208,1206,1206,1206,1210,1210,
1208,1206,1210,1210,1208,1206
1204 GOSUB 1146:Y=19:X=75:RETURN
1206 GOSUB 1146:Y=Y-1:X=75:RETURN
1208 GOSUB 1146:Y=Y-3:X=75:RETURN
1210 RETURN
1212 '
1214 '          MOVE DOWN
1216 '
1218 ON Y GOTO 1220,1220,1220,1220,1222,1226,1226,1220,1220,1220,1222,1226,1226,
1220,1222,1226,1226,1220,1224
1220 GOSUB 1146:Y=Y+1:X=75:RETURN
1222 GOSUB 1146:Y=Y+3:X=75:RETURN
1224 GOSUB 1146:Y=1:X=75
1226 RETURN
1228 '
```

```
1230 '          AXES
1232 '
1234 NX=INT((MAX-MIX)/XT):NY=INT((MAY-MIY)/YT):MX=INT(XL/XT):MY=INT(YL/YT)
1236 XG=462-(422 MOD NX):YG=1+(159 MOD NY):LINE@(40,160)-(XG,YG),PSET,7,B
1238 FOR KY=1 TO NY:YL1=FNY(KY):IF KY MOD MY=0 THEN LINE(40,YL1)-(46,YL1),PSET,7
 ELSE LINE(40,YL1)-(43,YL1),PSET,7
1240 NEXT KY
1242 FOR KX=1 TO NX :XL1=FNX(KX): IF KX MOD MX=0 THEN LINE(XL1,157)-(XL1,160),PS
ET,7 ELSE LINE(XL1,158)-(XL1,160),PSET,7
1244 NEXT KX
1246 IF DG=0 THEN GO TO 1254
1248 FOR KX=MX TO NX STEP MX:XL1=FNX(KX):FOR KY=MY TO NY STEP MY:YL1=FNY(KY)
1250  PSET(XL1,YL1,7)
1252 NEXT KY,KX
1254 '
1256 '      LABEL
1258 '
1260 AY0=162:AY1=166
1262 FOR J=MIX TO MAX STEP XL:IF ABS(J)<.00001 THEN J=0
1264 IF MID$(STR$(J),1)="" THEN XL$=MID$(STR$(J),2) ELSE XL$=STR$(J)
1266 CN=LEN(XL$):CND=CN¥2:XL1=FNPX1(J)
1268 FOR I=0 TO CN-1
1270 AX0=XL1-7*CND+7*I:AX1=AX0+4
1272 CX$=MID$(XL$,I+1):CX=ASC(CX$):CNX=CX-44:IF CNX<0 GO TO 1276
1274 ON CNX GOSUB 1294,1296,1318,1298,1300,1302,1304,1306,1308,1310,1312,1314,13
16
1276 NEXT I,J
1278 FOR J=MIY TO MAY STEP YL:IF ABS(J)<.00001 THEN J=0
1280 IF MID$(STR$(J),1)="" THEN YL$=MID$(STR$(J),2) ELSE YL$=STR$(J)
1282 CN1=LEN(YL$):YL1=FNPY1(J)
1284 FOR I=0 TO CN1-1
1286 AY0=YL1-4:AY1=YL1:AX0=38-7*CN1+7*I:AX1=AX0+4
1288 CY$=MID$(YL$,I+1):CY=ASC(CY$):CNY=CY-44:IF CNY<0 GO TO 1292
1290 ON CNY GOSUB 1294,1296,1318,1298,1300,1302,1304,1306,1308,1310,1312,1314,13
16
1292 NEXT I,J:RETURN
1294 PUT@(AX0,AY0)-(AX1,AY1),A0%,PSET,7:RETURN
1296 PUT@(AX0,AY0)-(AX1,AY1),A1%,PSET,7:RETURN
1298 PUT@(AX0,AY0)-(AX1,AY1),A2%,PSET,7:RETURN
1300 PUT@(AX0,AY0)-(AX1,AY1),A3%,PSET,7:RETURN
1302 PUT@(AX0,AY0)-(AX1,AY1),A4%,PSET,7:RETURN
1304 PUT@(AX0,AY0)-(AX1,AY1),A5%,PSET,7:RETURN
1306 PUT@(AX0,AY0)-(AX1,AY1),A6%,PSET,7:RETURN
1308 PUT@(AX0,AY0)-(AX1,AY1),A7%,PSET,7:RETURN
1310 PUT@(AX0,AY0)-(AX1,AY1),A8%,PSET,7:RETURN
1312 PUT@(AX0,AY0)-(AX1,AY1),A9%,PSET,7:RETURN
1314 PUT@(AX0,AY0)-(AX1,AY1),A10%,PSET,7:RETURN
1316 PUT@(AX0,AY0)-(AX1,AY1),A11%,PSET,7
1318 RETURN
1320 '
1322 '          GCURSOR
1324 '
1326 PRINT
1328 PRINT:LOCATE0,24:PRINT "[ 1: Set  2: Reset  3: End  ]";
1330 SYMBOL(256,192),"Hit any Key",1,1,5
1332 GC$=INPUT$(1):IF GC$<"1" OR GC$>"3" THEN 1332ELSE GC=VAL(GC$)
1334 IF GC=3 THEN RETURN
1336 ON GC GOSUB 1338,1350:GO TO 1328
1338 '
1340 '          SET
1342 '
1344 SYMBOL(256,192),"Hit any Key",1,1,0:LOCATE32,24:INPUT "Text --> ",TX$
1346 PRINT"Set the one coordianate";
1348 GCURSOR(GX0,GY0),(GX,GY),6:SYMBOL(GX,GY),TX$,SI,SI,7,AN,PSET:TX=LEN(TX$):GX
0=GX-8*TX*(AN=0)+8*TX*(AN=2):GY0=GY+8*TX*(AN=1)-8*TX*(AN=3):RETURN
1350 '
1352 '          RESET
1354 '
1356 SYMBOL(256,192),"Hit any Key",1,1,0
1358 GCURSOR(GX0,GY0),(GX,GY),6:SYMBOL(GX,GY),CHR$(&H87),SI,SI,0,AN,PSET:GX0=GX-
8*(AN=0)+8*(AN=3):GY0=GY-8*(AN=2)+8*(AN=4)
1360 RETURN
1362 '
1364 '          DATA
1366 '
1368 PRINT
1370 COLOR 4:PRINT " NO:       X=           Y=    ";:COLOR 7:N$=STR$(N)
1372 SYMBOL(504,192),"Hit E(End)",1,1,5
1374 SYMBOL(32,192),N$,1,1,7:SYMBOL(112,192),DX$(N),1,1,7:SYMBOL(208,192),DY$(N)
,1,1,7
```

Auto Plot BASIC プログラム・リスト

```
1376 LOCATE XD,24:GOSUB 1408:DA$=INPUT$(1):DA=ASC(DA$)
1378 IF (DA$<="9" AND DA$>="0") OR (DA$="."OR (DA$="-" OR DA$=CHR$(&H20))) THEN
SYMBOL(8*XD,192),CHR$(&H87),1,1,0:SYMBOL(8*XD,192),DA$,1,1:IXD=(13-XD)*(XD<20)+(
20-XD)*(XD>20):DT$(N,IXD)=DA$:GOSUB 1416:GO TO 1376
1380 IF DA>=28 AND DA<=31 THEN SYMBOL(8*XD,192),CHR$(&H87),1,1,,XOR:ON DA-27 GO
SUB 1416,1430,1444,1454:GO TO 1376
1382 IF DA=69 THEN SYMBOL(8*XD,192),CHR$(&H87),1,1,,,XOR:GOSUB 1386:LOCATE0,24:P
RINT:RETURN
1384 GOSUB 1414:GO TO 1376
1386 '
1388 '          GET CHARACTER
1390 '
1392 DX$(N)="":FOR I=1 TO 5:DX$(N)=DX$(N)+DT$(N,I):NEXT:DX(N)=VAL(DX$(N))
1394 IF DX(N)>MAX THEN DX(N)=MAX
1396 IF DX(N)<MIX THEN DX(N)=MIX
1398 DY$(N)="":FOR I=6 TO 10:DY$(N)=DY$(N)+DT$(N,I):NEXT:DY(N)=VAL(DY$(N))
1400 IF DY(N)>MAY THEN DY(N)=MAY
1402 IF DY(N)<MIY THEN DY(N)=MIY
1404 LINE(32,192)-(64,199),PSET,0,BF:LINE(112,192)-(152,199),PSET,0,BF:LINE(208,
192)-(248,199),PSET,0,BF
1406 RETURN
1408 '
1410 '          PRINT(2)
1412 '
1414 SYMBOL(8*XD,192),CHR$(&H87),1,1,,,XOR:RETURN
1416 '
1418 '          MOVE RIGHT(2)
1420 '
1422 IF XD=18 THEN XD=26:RETURN
1424 IF XD=30 THEN XD=14:GOSUB 1454:RETURN
1426 IF (XD>=14 AND XD<18) OR (XD>=26 AND XD<30) THEN XD=XD+1
1428 RETURN
1430 '
1432 '          MOVE LEFT(2)
1434 '
1436 IF (XD>14 AND XD<=18) OR (XD>26AND XD<=30) THEN XD=XD-1:RETURN
1438 IF XD=14 THEN XD=30:GOSUB 1444:RETURN
1440 IF XD=26 THEN XD=18
1442 RETURN
1444 '
1446 '          BACK
1448 '
1450 GOSUB 1386:N=N-1-(N=1):N$=STR$(N)
1452 SYMBOL(32,192),N$,1,1,7:SYMBOL(112,192),DX$(N),1,1,7:SYMBOL(208,192),DY$(N)
,1,1,7:XD=14:RETURN
1454 '
1456 '          NEXT
1458 '
1460 GOSUB 1386:N=N+1:N$=STR$(N)
1462 SYMBOL(32,192),N$,1,1,7:SYMBOL(112,192),DX$(N),1,1,7:SYMBOL(208,192),DY$(N)
,1,1,7:XD=14:RETURN
1464 '
1466 '          PLOT
1468 '
1470 FOR I=1 TO PT:PX(I)=FNPX(I):PY(I)=FNPY(I):NEXT
1472 '
1474 '          SORT
1476 '
1478 IK=PT
1480 IK=INT((IK+1)/2):S=PT-IK
1482 FOR I=1 TO S:J=I:IF PX(I)<=PX(I+IK) THEN 1492
1484 SWAP PX(J),PX(J+IK):SWAP PY(J),PY(J+IK)
1486 IF (J-IK)<1 THEN 1492
1488 IF PX(J-IK)<=PX(J) THEN 1492
1490 J=J-IK:GO TO 1484:NEXT
1492 NEXT
1494 IF IK<>1 THEN 1480
1496 '
1498 '          PLOT SUB
1500 '
1502 ON LT GOSUB 1506,1508,1518
1504 RETURN
1506 FOR IP=1 TO PT-1:CONNECT(PX(IP),PY(IP))-(PX(IP+1),PY(IP+1)),7:NEXT:RETURN
1508 FOR IP=1 TO PT-1
1510 IF FNM(IP)<=1 AND FNM(IP)>=-1 THEN FOR KP=PX(IP) TO PX(IP+1)-5 STEP 10:CONN
ECT(KP,FNDPY(KP,IP))-(KP+3,FNDPY(KP+3,IP)),7:NEXT KP
1512 IF FNM(IP)>1 THEN FOR KP=PY(IP) TO PY(IP+1)-5 STEP 10:CONNECT(FNDPX(KP,IP),
KP)-(FNDPX(KP+3,IP),KP+3),7:NEXT KP
1514 IF FNM(IP)<-1 THEN FOR KP=PY(IP) TO PY(IP+1)+5 STEP -10:CONNECT(FNDPX(KP,IP
```

```
),KP)-(FNDPX(KP-3,IP),KP-3),7:NEXT KP
1516 NEXT IP:RETURN
1518 FOR IP=1 TO PT-1
1520 IF FNM(IP)<=1 AND FNM(IP)>=-1 THEN FOR KP=PX(IP) TO PX(IP+1)-5 STEP 15:CONN
ECT(KP,FNDPY(KP,IP))-(KP+7,FNDPY(KP+7,IP)),7:NEXT KP
1522 IF FNM(IP)>1 THEN FOR KP=PY(IP) TO PY(IP+1)-5 STEP 15:CONNECT(FNDPX(KP,IP),
KP)-(FNDPX(KP+7,IP),KP+7),7:NEXT KP
1524 IF FNM(IP)<-1 THEN FOR KP=PY(IP) TO PY(IP+1)+5 STEP -15:CONNECT(FNDPX(KP,IP
),KP)-(FNDPX(KP-7,IP),KP-7),7:NEXT KP
1526 NEXT IP:RETURN
1528 '
1530 '          SUB HARD COPY ( 2 )
1532 '          DOT REGION 472*179
1534 '
1536 PRINT #2,CHR$(&H1B);"I";CHR$(24);
1538 PRINT #2,CHR$(&H1B);"2";
1540 FOR I=0 TO 472 STEP 8
1542 PRINT #2,CHR$(&H1B);"K";
1544 PRINT #2,CHR$(&H68);CHR$(&H1);
1546 FOR J=179 TO 0 STEP -1
1548 GET@(I,J)-(I+7,J),G%,G,7
1550 DO=G%(0)¥256 AND 255
1552 PRINT #2,CHR$(DO);CHR$(DO);
1554 NEXT J
1556 PRINT #2,CHR$(&HD);CHR$(&HA);
1558 NEXT I:RETURN
1560 '
1562 '          CLS
1564 '
1566 LINE@(0,0)-(479,179),PSET,0,BF:RETURN 1104
1568 '
1570 '          END
1572 '
1574 WIDTH 80,25
1576 END
```



FM-8

フローチャート・ジェネレータ

■鱧谷　憲

できあがったプログラムをフローチャートにして整理すれば，プログラムの内容の表示，改良，プログラムの追加などに非常に便利です．また，他人から貰ったプログラムの内容を調べたりするときにも，フローチャートがあれば役に立ちます．しかし，プログラムと1対1に対応するフローチャートを手で書くのは大変です．そこで，フローチャートを自動的に作成するプログラムを作ってみました．例としてフローチャート・ジェネレータ自身のメイン・プログラムをフローチャートにしてみました(図1)．

使い方

始めにフローチャートにしてみたいプログラムをASCIIでセーブします．コマンドは**SAVE〝プログラム名〃，A**です．フローチャート・ジェネレータを走らせ，先のテープをカセットにセットし，フローチャートにしたい始めの行，終りの行，プログラム名を順次入力し，もう一度，**RETURN**キーを押すとテープが回り出し，飛び先のチェックが行なわれます．

表示されている数字は行番号を表わしています．次に，Drawチェックが行なわれ，線がかち合っていないか，などの処理が行なわれます．さらに，**RETURN**キーを押し，テープをもう一度読ませると順次フローチャートがプリンタからでてきます．

プログラムについて

プリンタがない場合は，表示文字数を80桁にして，40行の〝LPT0：〃を〝SCRN：〃にします．プリンタ出力の部分は1文字でも間違うと図がずれてしまいます．7010行にスケールを付けました．1210行，その他の所の〝….02＋.02…〃を〝＋.04〃に変えると，Drawチェックが動きません．

このプログラムはFM-8のプログラムを，すべてのケースでフローチャートに変換します．ただし，図2のような場合は2番目の**NEXT**文については線を引きません．また，プログラムとして走らないものについてはエラーストップすることもあります．

注　意

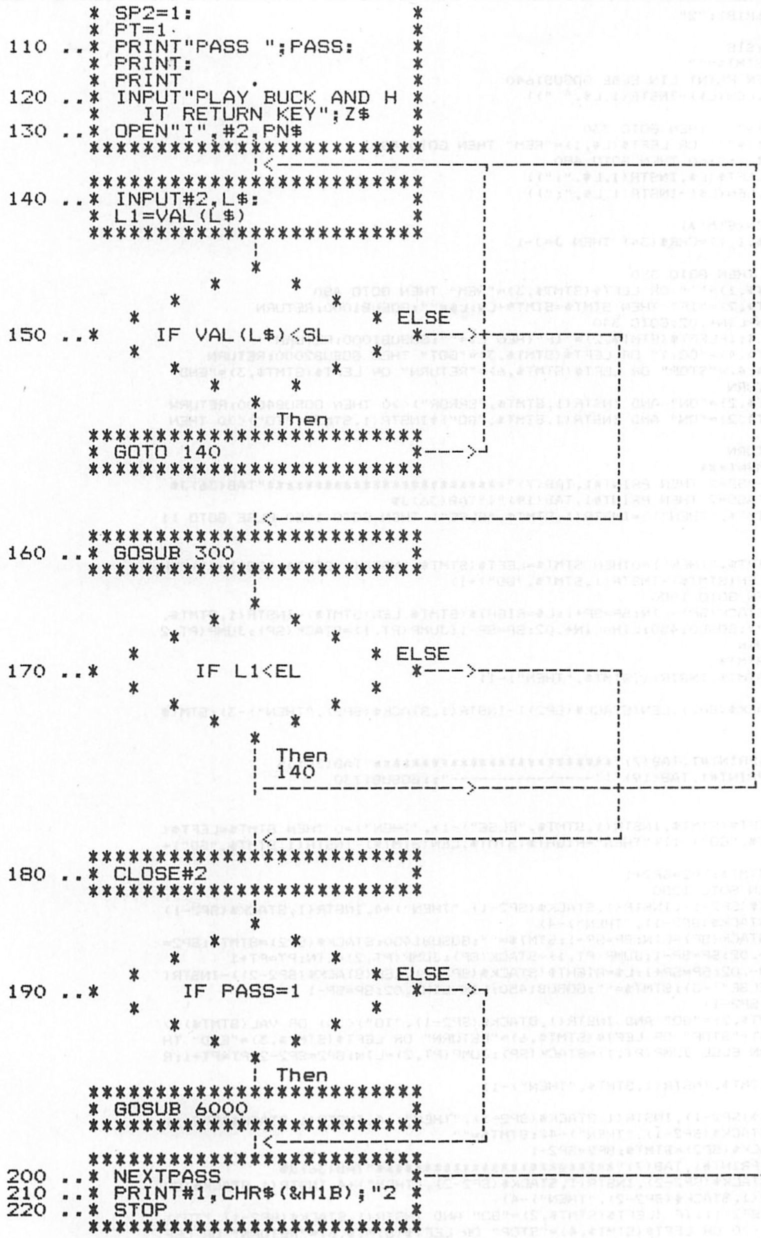
プログラムが非常に長い場合は，Drawチェックに時間がかかりますが，半分ずつに分割すればおよそ4分の1になります．適当なサブルーチンごとに処理されることをお勧めします．

表1

プログラムの内容	行　番　号
MAIN	10～　220
文解析	300～　530
IF文	1000～1830
GOTO文	2000～2290
STOP文	3000～3080
ONERROR文	4000～4070
ON GOTO文	5000～5300
Drawチェック	6000～6530

図1

```
          ***************************
  10   ..* 'MAIN PROGRAM            *
  20   ..* CLEAR10000               *
  30   ..* DIMJUMP(300,3),STACK(    *
         *   50),STACK$(50),LN(5    *
         *   0),JT(100)             *
  40   ..* OPEN"O",#1,"LPTO:"       *
  50   ..* PRINT#1,CHR$(&H1B);"O    *
  60   ..* INPUT"START LINE NO "    *
         *   ,SL                    *
  70   ..* INPUT"END LINE NO ",E    *
  80   ..* INPUT"PROGRAM NAME ",    *
         *   PN$                    *
  90   ..* FOR PASS=1TO2            *-----------------------------┐
 100   ..* SP=10:                   *                             ¦
         * FL=0:                    *                             ¦
```



フローチャート・ジェネレータ BASICリスト

```
10 'MAIN PROGRAM
20 CLEAR10000
30   DIMJUMP(300,3),STACK(50),STACK$(50),LN(50),JT(100)
40   OPEN"O",#1,"LPT0:"
50 PRINT#1,CHR$(&H1B);"O"
60   INPUT"START LINE NO ",SL
70   INPUT"END LINE NO ",EL
80   INPUT"PROGRAM NAME ",PN$
90 FOR PASS=1TO2
100 SP=10:FL=0:SP2=1:PT=1
110 PRINT"PASS ";PASS:PRINT:PRINT
120 INPUT"PLAY BUCK AND HIT RETURN KEY";Z$
130 OPEN"I",#2,PN$
140 INPUT#2,L$:L1=VAL(LEFT$(L$,INSTR(L$," ")))
150 IF L1<SL THEN GOTO 140
160 GOSUB 300
170 IF L1<EL THEN 140
180 CLOSE#2
190 IF PASS=1 THEN GOSUB 6000
200 NEXTPASS
```

フローチャート・ジェネレータ BASICリスト

```
210 PRINT#1,CHR$(&H1B);"2"
220 STOP
300  'SYNTAX ANALYSIS
310  LIN=VAL(L$):STMT$=""
320  IF PASS=1 THEN PRINT LIN ELSE GOSUB1640
330  L$=RIGHT$(L$,LEN(L$)-INSTR(1,L$," "))
340  STMT$=""
350 IF LEFT$(L$,1)=" " THEN GOTO 330
360  IF LEFT$(L$,1)="'" OR LEFT$(L$,1)="REM" THEN GOTO 480
370  IF INSTR(1,L$,":")=0 THEN GOTO 480
380  STMT$=STMT$+LEFT$(L$,INSTR(1,L$,":"))
390  L$=RIGHT$(L$,LEN(L$)-INSTR(1,L$,":"))
400  J=0
410  FOR I=1 TO LEN(STMT$)
420  IF MID$(STMT$,I,1)=CHR$(34) THEN J=J+1
430  NEXTI
440  IF J MOD 2=1 THEN GOTO 350
450  IF LEFT$(STMT$,1)="'" OR LEFT$(STMT$,3)="REM" THEN GOTO 480
460  IF LEFT$(STMT$,2)="IF" THEN STMT$=STMT$+L$:L$="":GOSUB1000:RETURN
470  GOSUB 490:LIN=LIN+.02:GOTO 340
480  STMT$=STMT$+L$:IFLEFT$(STMT$,2)="IF"THEN L$="":GOSUB1000:RETURN
490  IF LEFT$(STMT$,4)="GO T" OR LEFT$(STMT$,3)="GOT" THEN GOSUB2000:RETURN
500  IFLEFT$(STMT$,4)="STOP" OR LEFT$(STMT$,6)="RETURN" OR LEFT$(STMT$,3)="END"
THEN GOSUB3000:RETURN
510  IF LEFT$(STMT$,2)="ON" AND INSTR(1,STMT$,"ERROR")<>0 THEN GOSUB4000:RETURN
520  IF LEFT$(STMT$,2)="ON" AND INSTR(1,STMT$,"GO")*INSTR(1,STMT$,"TO")<>0 THEN
GOSUB5000:RETURN
530  GOSUB2090:RETURN
1000 '*** IF STAEMENT***
1010 IF FL=2 AND PASS=2 THEN PRINT#1,TAB(7)"**************************"TAB(36)J$
1020 IF FL=2 AND PASS=2 THEN PRINT#1,TAB(19)"|"TAB(36)J$
1030 IF INSTR(1,STMT$,"THEN")>=INSTR(1,STMT$,"ELSE") THEN GOTO 1050 ELSE GOTO 11
70
1040 RETURN
1050 IF INSTR(1,STMT$,"THEN")=0THEN STMT$=LEFT$(STMT$,INSTR(1,STMT$,"GO")-1)+"TH
EN"+RIGHT$(STMT$,LEN(STMT$)-INSTR(1,STMT$,"GO")+1)
1060 IF PASS=2 THEN GOTO 1080
1070 LIN=LIN+.02:STACK(SP)=LIN:SP=SP+1:L$=RIGHT$(STMT$,LEN(STMT$)-INSTR(1,STMT$,
"THEN")-3):STMT$="":GOSUB1450:LIN=LIN+.02:SP=SP-1:JUMP(PT,1)=STACK(SP):JUMP(PT,2
)=LIN:PT=PT+1:RETURN
1080 STACK$(SP2)=STMT$
1090 STMT$=LEFT$(STMT$,INSTR(1,STMT$,"THEN")-1)
1100 GOSUB1460
1110 L$=RIGHT$(STACK$(SP2),LEN(STACK$(SP2))-INSTR(1,STACK$(SP2),"THEN")-3):STMT$
=""
1120 GOSUB1450
1130 IF FL=2 THEN PRINT#1,TAB(7)"**************************"TAB(36)J$
1140 LIN=LIN+.02:PRINT#1,TAB(19)"|<----------------";:GOSUB1730
1150 FL=1
1160 RETURN
1170 IF INSTR(1,LEFT$(STMT$,INSTR(1,STMT$,"ELSE")-1),"THEN")=0 THEN STMT$=LEFT$(
STMT$,INSTR(1,STMT$,"GO")-1)+"THEN"+RIGHT$(STMT$,LEN(STMT$)-INSTR(1,STMT$,"GO")+
1)
1180 STACK$(SP2)=STMT$:SP2=SP2+1
1190 IF PASS=2 THEN GOTO 1250
1200 L$=MID$(STACK$(SP2-1),INSTR(1,STACK$(SP2-1),"THEN")+4,INSTR(1,STACK$(SP2-1)
,"ELSE")-INSTR(1,STACK$(SP2-1),"THEN")-4)
1210 LIN=LIN+.02:STACK(SP)=LIN:SP=SP+1:STMT$="":GOSUB1450:STACK$(SP2)=STMT$:SP2=
SP2+1:LIN=LIN+.02+.02:SP=SP-1:JUMP(PT,1)=STACK(SP):JUMP(PT,2)=LIN:PT=PT+1
1220 STACK(SP)=LIN-.02:SP=SP+1:L$=RIGHT$(STACK$(SP2-2),LEN(STACK$(SP2-2))-INSTR(
1,STACK$(SP2-2),"ELSE")-3):STMT$="":GOSUB1450:LIN=LIN+.02:SP=SP-1
1230 STMT$=STACK$(SP2-1)
1240 IF (LEFT$(STMT$,2)="GO" AND INSTR(1,STACK$(SP2-1),"TO")<>0) OR VAL(STMT$)<>
0 OR LEFT$(STMT$,4)="STOP" OR LEFT$(STMT$,6)="RETURN" OR LEFT$(STMT$,3)="END" TH
EN SP2=SP2-2:RETURN ELSE JUMP(PT,1)=STACK(SP):JUMP(PT,2)=LIN:SP2=SP2-2:PT=PT+1:R
ETURN
1250 STMT$=LEFT$(STMT$,INSTR(1,STMT$,"THEN")-1)
1260 GOSUB1460
1270 L$=MID$(STACK$(SP2-1),INSTR(1,STACK$(SP2-1),"THEN")+4,INSTR(1,STACK$(SP2-1)
,"ELSE")-INSTR(1,STACK$(SP2-1),"THEN")-4):STMT$=""
1280 GOSUB1450:STACK$(SP2)=STMT$:SP2=SP2+1
1290 IF FL=2 THEN PRINT#1,TAB(7)"**************************"TAB(36)J$
1300 'STMT$=MID$(STACK$(SP2-2),INSTR(1,STACK$(SP2-2),"THEN")+4,INSTR(1,STACK$(SP
2-2),"ELSE")-INSTR(1,STACK$(SP2-2),"THEN")-4)
1310 STMT$=STACK$(SP2-1):IF (LEFT$(STMT$,2)="GO" AND INSTR(1,STACK$(SP2-1),"TO")
<>0) OR VAL(STMT$)<>0 OR LEFT$(STMT$,4)="STOP" OR LEFT$(STMT$,6)="RETURN" OR LEF
T$(STMT$,3)="END" THEN FL=0:LIN=LIN+.02:GOTO1340
1320 FL=1
1330 LIN=LIN+.02:PRINT#1,TAB(19)"|---------------->";:GOSUB1730
1340 PRINT#1,TAB(36)J$
1350 LIN=LIN+.02:PRINT#1,TAB(19)"|<----------------";:GOSUB1730
1360 STMT$=STACK$(SP2-2)
1370 L$=RIGHT$(STMT$,LEN(STMT$)-INSTR(1,STMT$,"ELSE")-3):STMT$=""
1380 GOSUB1450
1390 IF FL=2 THEN PRINT#1,TAB(7)"**************************"TAB(36)J$
1400 STMT$=STACK$(SP2-1)
1410 IF (LEFT$(STMT$,2)="GO" AND INSTR(1,STACK$(SP2-1),"TO")<>0) OR VAL(STMT$)<>
0 OR LEFT$(STMT$,4)="STOP" OR LEFT$(STMT$,6)="RETURN" OR LEFT$(STMT$,3)="END" TH
EN FL=0:LIN=LIN+.02:GOTO1430
1420 LIN=LIN+.02:PRINT#1,TAB(19)"|<----------------";:GOSUB1730
1430 FL=1:SP2=SP2-2
1440 RETURN
1450 IF VAL(L$)<>0THEN STMT$=","+L$:GOSUB5040:RETURN:ELSEGOSUB 350:RETURN
1460 '********* IF ***********
1470 PRINT#1,TAB(7);"             *                  ";TAB(36)J$
1480 PRINT#1,TAB(7);"          *      *              ";TAB(36)J$
1490 PRINT#1,TAB(7);"       *             *          ";:PRINT#1,J$
1500 PRINT#1,TAB(7);"   *                   * ELSE ";TAB(36)J$
1510 IF LIN=INT(LIN) THENPRINT#1,USING"#### ..";LIN;
1520 LIN=LIN+.02:PRINT#1, TAB(7);"*";TAB(19-LEN(LEFT$(STMT$,17))/2);LEFT$(STMT$,
17)TAB(31)"*--->";:GOSUB1730
1530 WHILELEN(STMT$)>18
1540 STMT$=RIGHT$(STMT$,LEN(STMT$)-17)
```

```
1550 PRINT#1,TAB(11)LEFT$(STMT$,17);TAB(36)J$
1560 WEND
1570 PRINT#1,TAB(7);"   *              *          ";TAB(36)J$
1580 PRINT#1,TAB(7);"      *           *          ";TAB(36)J$
1590 PRINT#1,TAB(7);"         *     *             ";TAB(36)J$
1600 PRINT#1,TAB(7);"            *                ";TAB(36)J$
1610 PRINT#1,TAB(7);"            ¦ Then           ";TAB(36)J$
1620 FL=1
1630 RETURN
1640 '****** LINE DRAW ******
1650 I=1
1660 IF JT(I)=0 THEN 1700
1670 IF JT(I)=LIN THEN 1720
1680 I=I+1
1690 GOTO 1660
1700 IF STMT$<>"" THEN PRINT#1,TAB(36)J$:RETURN
1710 IF STMT$=""THEN RETURN
1720 IF FL=2 AND LIN=INT(LIN) THEN PRINT#1,TAB(7)"**************************"TAB(
36)J$
1730 I=1:J=0:J$=STRING$(42," "):K$=STRING$(42,"-")
1740 WHILE JUMP(I,1)<>0
1745 IF JUMP(I,3)=0 THEN GOTO 1790
1750 IF JUMP(I,1)<LIN+.01 AND JUMP(I,2)>LIN+.01 THEN MID$(J$,JUMP(I,3),1)="¦":MI
D$(K$,JUMP(I,3),1)="+"
1760 IF (JUMP(I,1)=LIN OR JUMP(I,2)=LIN) AND    JUMP(I,3)>J THEN J=JUMP(I,3)
1770 IF JUMP(I,1)=LIN THEN MID$(K$,JUMP(I,3),1)=CHR$(&H99)
1780 IF JUMP(I,2)=LIN THEN MID$(K$,JUMP(I,3),1)=CHR$(&H9B)
1790 I=I+1
1800 WEND
1810 IF LIN=INT(LIN) THEN PRINT#1,TAB(19)"¦<----------------";:FL=1
1820 PRINT#1,TAB(36)LEFT$(K$,J);RIGHT$(J$,LEN(J$)-J)
1830 RETURN
2000 '*** GO TO STATEMENT ***
2010 IF PASS=1 THEN LIN=LIN+.02:JUMP(PT,1)=LIN:JUMP(PT,2)=VAL(RIGHT$(STMT$,LEN(S
TMT$)-INSTR(1,STMT$,"TO")-1)):PT=PT+1:RETURN
2020 IF FL<>2THEN PRINT#1,TAB(7)"**************************"TAB(36)J$
2030 IF LIN=INT(LIN) THENPRINT#1,USING"#### ..";LIN;
2040 LIN=LIN+.02:PRINT#1, TAB(7);"* "STMT$TAB(31)"*--->";:GOSUB1730
2050 PRINT#1,TAB(7)"**************************"TAB(36)J$
2060 PRINT#1,TAB(36)J$
2070 PRINT#1,TAB(36)J$
2080 FL=0:RETURN
2090 'FOR;NEXT;WHILE;WEND CHECK
2100 IFPASS=2 THEN GOTO 2180
2110 IF LEFT$(STMT$,3)="FOR" OR LEFT$(STMT$,5)="WHILE" THEN STACK(SP)=LIN+.02:SP
=SP+1:LIN=LIN+.02:RETURN
2120 IF LEFT$(STMT$,4)="WEND" THEN SP=SP-1:JUMP(PT,1)=STACK(SP):JUMP(PT,2)=LIN+.
02:PT=PT+1:LIN=LIN+.02:RETURN
2130 IFLEFT$(STMT$,4)<>"NEXT"THEN RETURN
2140 J=0:FORI=1TOLEN(STMT$)
2150 IFMID$(STMT$,I,1)=","THEN J=J+1
2160 NEXTI
2170 LIN=LIN+.02:FORI=1TOJ+1:SP=SP-1:JUMP(PT,1)=STACK(SP):JUMP(PT,2)=LIN:PT=PT+1
:NEXTI:RETURN
2180 '** NORMAL STATEMENT
2190 IF FL<>2THEN PRINT#1,TAB(7)"**************************"TAB(36)J$
2200 IF LIN=INT(LIN) THENPRINT#1,USING"#### ..";LIN;
2210 PRINT#1, TAB(7);"* "LEFT$(STMT$,21)TAB(31)"*";
2220 IFLEFT$(STMT$,3)="FOR" OR LEFT$(STMT$,4)="NEXT" OR LEFT$(STMT$,5)="WHILE" O
R LEFT$(STMT$,4)="WEND" THEN LIN=LIN+.02:PRINT#1,"----";:GOSUB1730:ELSE PRINT#1,
TAB(36)J$
2230 IF STMT$=""THEN GOTO 2290
2240 IF LEN(STMT$)>2 THEN STMT$=RIGHT$(STMT$,LEN(STMT$)-2)
2250 WHILE LEN(STMT$)>20
2260 STMT$=RIGHT$(STMT$,LEN(STMT$)-19)
2270 PRINT#1,TAB(7)"*    "LEFT$(STMT$,19)TAB(31)"*"TAB(36)J$
2280 WEND
2290 FL=2:RETURN
3000 '** STOP RETURN END STATEMENT***
3010 IF PASS=1THEN RETURN
3020 IF FL<>2THEN PRINT#1,TAB(7)"**************************"TAB(36)J$
3030 IF LIN=INT(LIN) THENPRINT#1,USING"#### ..";LIN;
3040 PRINT#1, TAB(7);"* "STMT$TAB(31)"*"TAB(36)J$
3050 PRINT#1,TAB(7)"**************************"TAB(36)J$
3060 PRINT#1,TAB(36)J$:PRINT#1,TAB(36)J$:PRINT#1,TAB(36)J$
3070 PRINT#1,TAB(36)J$:PRINT#1,TAB(36)J$:PRINT#1,TAB(36)J$
3080 FL=0:RETURN
4000 '*** ON ERROR ***
4030 IF PASS=2 THEN GOSUB5090
4040 STMT$=RIGHT$(STMT$,LEN(STMT$)-INSTR(1,STMT$,"ERROR")-4)
4050 GOSUB2000
4070 RETURN
5000 '*** ON GOTO ***
5010 IF PASS=2 THEN GOSUB5090
5020 STMT$=RIGHT$(STMT$,LEN(STMT$)-INSTR(1,STMT$,"TO")-1)
5030 LIN=LIN+.02:IF PASS=1 THEN JUMP(PT,1)=LIN:JUMP(PT,2)=VAL(STMT$):PT=PT+1:ELS
E PRINT#1,TAB(19)"¦";VAL(STMT$)TAB(36)J$:PRINT#1,TAB(19)"¦---------------->";:GOS
UB1730
5040 WHILE INSTR(1,STMT$,",")<>0
5050 STMT$=RIGHT$(STMT$,LEN(STMT$)-INSTR(1,STMT$,","))
5060 LIN=LIN+.02:IF PASS=1 THEN JUMP(PT,1)=LIN:JUMP(PT,2)=VAL(STMT$):PT=PT+1:ELS
EPRINT#1,TAB(19)"¦";VAL(STMT$)TAB(36)J$:PRINT#1,TAB(19)"¦---------------->";:GOSU
B1730
5070 WEND
5080 IF PASS=1 THEN RETURN ELSE PRINT#1,TAB(36)J$:PRINT#1,TAB(36)J$:FL=0:RETURN
5090 '*** ON ***
5100 IF FL=2 THEN PRINT#1,TAB(7)"**************************"TAB(36)J$
5110 IF FL=2 THENPRINT#1,TAB(19)"¦"TAB(36)J$
5120 PRINT#1,TAB(7);"            *                ";TAB(36)J$
5130 PRINT#1,TAB(7);"         *     *             ";TAB(36)J$
5140 PRINT#1,TAB(7);"      *           *          "TAB(36)J$
5150 PRINT#1,TAB(7);"   *                 *       ";TAB(36)J$
5160 IF LIN=INT(LIN) THENPRINT#1,USING"#### ..";LIN;
```

フローチャート・ジェネレータ BASICリスト

```
5170 STACK$(SP2)=STMT$:SP2=SP2+1
5180 PRINT#1, TAB(7);"*";TAB(19-LEN(LEFT$(STMT$,17))/2);LEFT$(STMT$,17)TAB(31)"*
"TAB(36)J$
5190 WHILELEN(STMT$)>18
5200 STMT$=RIGHT$(STMT$,LEN(STMT$)-17)
5210 PRINT#1,TAB(11)LEFT$(STMT$,17);TAB(36)J$
5220 WEND
5230 SP2=SP2-1:STMT$=STACK$(SP2)
5240 PRINT#1,TAB(7);"   *                      *       ";TAB(36)J$
5250 PRINT#1,TAB(7):"       *               *          ";TAB(36)J$
5260 PRINT#1,TAB(7);"           *       *              ";TAB(36)J$
5270 PRINT#1,TAB(7);"               *                  ";TAB(36)J$
5280 PRINT#1,TAB(7);"               !";TAB(36)J$
5290 FL=1
5300 RETURN
6000 '*** DRAW CHECK***
6010 PRINT:PRINT:PRINT"DRAW CHECK"
6020 I=1:MAX=0:PT=1
6030 WHILE JUMP(I,1)<>0
6040 IF JUMP(I,1)>JUMP(I,2) THEN SWAP JUMP(I,1),JUMP(I,2)
6041 IF INT(JUMP(I,1))=JUMP(I,1) THEN JT(PT)=JUMP(I,1):PT=PT+1
6042 IF INT(JUMP(I,2))=JUMP(I,2) THEN JT(PT)=JUMP(I,2):PT=PT+1
6050 I=I+1
6060 WEND
6070 PRINT"CHECK 1   OK!"
6080 PRINT" CHECK 2"
6090 MIN=-1
6100 LIN=64000!
6110 GOSUB6260
6120 IF LIN=64000!THENGOTO6210
6121 IF INT(LIN)=LIN THEN GOTO6130
6122 IF JUMP(PT,1)=LIN THEN GOSUB6350:GOTO 6190
6123 IF JUMP(PT,2)=LIN THEN GOSUB6440:PT=PT+1:GOTO6123:ELSE GOTO6190
6130 PT=1
6140 WHILE JUMP(PT,2)<>0
6150 IF JUMP(PT,1)=LIN THEN GOSUB6350
6160 IF JUMP(PT,2)=LIN THEN GOSUB6440
6170 PT=PT+1
6180 WEND
6190 MIN=LIN
6200 GOTO6100
6210 PRINT"CHECH 2   OK!"
6220 IF MAX<21 THEN I=1:WHILE JUMP(I,1)<>0:JUMP(I,3)=(JUMP(I,3)-1)*INT(40/MAX)+1
:I=I+1:WEND:PRINT"CHECK 3   OK!"
6230 PRINT"DRAW LINE ALL OK"
6240 'I=1:WHILE JUMP(I,1)<>0:PRINT#1,I;TAB(5)JUMP(I,1)TAB(15)JUMP(I,2)TAB(25)JUM
P(I,3):I=I+1:WEND
6241 'I=1:WHILE JT(I)<>0:PRINT I;TAB(5)JT(I):I=I+1:WEND
6250 RETURN
6260 I=1
6270 WHILE JUMP(I,2)<>0
6280 IF JUMP(I,1)<=MIN THEN GOTO6300
6290 IF JUMP(I,1)<LIN THEN LIN=JUMP(I,1):PT=I
6300 IF JUMP(I,2)<=MIN THEN GOTO6320
6310 IF JUMP(I,2)<LIN THEN LIN=JUMP(I,2):PT=I
6320 I=I+1
6330 WEND
6340 RETURN
6350 L=0:FORK=1TO42
6360 IF LN(K)=0 THEN GOTO6380
6370 IF JUMP(LN(K),2)<JUMP(PT,2)  THEN 6375
6371 IF K=1 THEN GOTO6400
6372 IF LN(K) =0 THENK=L:GOTO6400
6373 IF LN(K)<>0 THENK=L+1:GOTO6400
6375 L=L+1
6380 NEXT K
6390 K=L+1
6400 J=K:WHILE LN(K)<>0 AND J<42:J=J+1:SWAP LN(J),LN(K):WEND
6410 LN(K)=PT
6420 'PRINT#1,LIN;:FORJ=1TO15:PRINT#1,TAB(J*4+4)LN(J);:NEXT J:PRINT#1
6430 RETURN
6440 FORJ=1TO42
6450 IF LN(J)=PT THEN GOTO6480
6460 NEXTJ
6470 JUMP(PT,3)=0:RETURN
6480 IFJ>40THEN JUMP(PT,3)=40 ELSE JUMP(PT,3)=J
6490 LN(J)=0
6500 IF MAX<J THEN MAX=J
6510 'PRINT#1,LIN;:FORJ=1TO15:PRINT#1,TAB(J*4+4)LN(J);:NEXT J:PRINT#1
6520 PRINT USING"####### -###### ###";JUMP(PT,1),JUMP(PT,2),JUMP(PT,3)
6530 RETURN
7000 '
7010 '+||||.||||+||||.||||+||||.||||+||||.||||+||||.||||+||||.||||+||||.||||+
7020 '0        10        20        30        40        50        60        70
```

FM-8 有象無象（うぞうむぞう）

■水島達彦

PART 1
ROM→RAM

FM-8は，64Kビット/チップのDRAMを採用しているのですが，ROM-BASICを使っているときには，半分の32Kバイトが眠ってしまいます．しかも，私の知っている限りROMの切り離しを，バンク・セレクトのようにソフトで行なうことはできません．

このままでは癪にさわるので**リスト1**のようなプログラムを組んでF-BASICインタープリタをRAM上に転送し，RAM上でF-BASICを走らせるものです．ただしROMの切り離しはユーザーが本体後部のディップ・スイッチを切り換えることで行ないます．

普通はあまり価値のあるプログラムではありませんが，私はF-BASICインタープリタ上にブレーク・ポイントを設定するために作っております．F-BASICは$8000～$87FFまでを使っていないため，フリーエリアは2Kバイト増えますが，ブートROMが違うために **RESET** および **STOP**，**RESET** を行なうと，ベルを鳴らし無限ループに落ちこみます．

リスト1のプログラムは6809のマシン語で，どのアドレスに置いても走ります．ここでは$7F00～のアドレスに入っていますが，ここは通常BASICの文字変数領域です．**EXEC &H7F00**で走らせ，しばらくするとベルが鳴りますから，ディップ・スイッチを切り変えてください．ディップ・スイッチは，ROM-BASICモード以外のどのモードでもかまいません．切り変えたら **STOP** を押します．これでRAM上のF-BASICがコールド・スタートします．

PART 2
REMARK→REMARK

FM-8は**REMARK**文を任意の色で表示させることができます．LIST処理のとき出力する文字列の中に，$11：DC1のコードがあると，次の1バイトの下位4ビットをカラーコードとして，その行の終りまでをその色で表示します．反転も可能ですが次の行に影響が出ることがあります．具体的に行なう操作を図1に示します．

行数が多くなるとこの操作はえらく時間を食うので，**リスト2**のプログラムを組んで自動的にやらせてしまいます．まず色をつけたい行を"行番号'@@A～"と書き替えます．2つの"@"があるアドレスには$0D：$11のコードが入ります．次の"A"はどんな文字でもよく，この文字のASCIIコードの下位4ビットが，カラーコードになります．以降はカラーで表示したい文字列を書きこんでください．終ったら**リスト2**のプログラムを走らせます．

リスト2も**6809**のマシン語で，アドレスに無関係に動きます．この場合は，**EXEC &H0500**で走らせます．アッという間に処理が終り**"Ready"**が表示されるので，LISTをとってみましょう．

行番号ずらつかない色のついたREMARK文がスクロール・アップしてゆき，非常に不気味です．なお，**リスト2**のプログラムは行の先頭以外の"'@@"や"REM"には反応しません．また$0500―$0510はF-BASICのキー入力用バッファですので，プログラムを**LOAD**した後に，190文字以上の文字列を入力すると，プログラムが破壊される可能性があります．

PART 3
LET'S GO MUSIC

リスト3はFM-8のカセット・ポートから音楽（らしきもの）を出力するものです．メインルーチンは6809のマシン語ですが，BASICのDATA文の形にしてあります（140行～490行）．これもどのアドレスでも動きます．アドレスを変更するときは**リスト3**のプログラム リスト中の（＊1）の数値を変えてください．

500行以降は演奏用のデータで，実行時にはメインルーチンの直後のアドレスから順に**POKE**してゆきます．データは音の長さ，音の高さの2つで1組になっており，ともに1バイトです．音の長さは20msec単位の時間です．これが0となると，終了とみなしBASICへ戻ります．

音の高さは表1の表のように各音階と対応しています．これを0にしておくと休符になります．

走らせ方：

まずプログラムをロードします．演奏データが入っていない場合は，音が出ないので，データを打ちこんでください．演奏データの最後に"DATA 256"を入れておいてください．音はカセット・ポートに出力されるので，カセットコードの赤いプラグの先に，アンプやラジカセをつないで音が出る状態にしてください．

"RUN"させるとメイン・ルーチンとデータをメモリ上に書き込みますから，10数秒待ってください．準備ができると**DATA SET READY**と表示されるので，何かキーを押してください．音楽が聞こえるはずです．

170行の**DATA**を"CE, FD, 03"，390行を"86, 81"または"86, 61"，440行を"86, 01"とすることにより，FM-8内のスピーカーから音を出すことができますが，めっちゃんこいやな音がします．もっともカセット・ポートから出る音も"美しい音"とはいいかねますが…．

PART 4
INTERVAL TIMER

F-BASICは，1秒単位でのINTERVAL割り込みを可能にしていますが，実は20msec 単位でのINTERVAL割り込みを設定することが可能です．

PART 3のプログラムは音の長さをこの機能で決めているのです．どのようにして設定するかというと，$01AC～$01AFの4バイトに20msec単位の値を**POKE**し，**EXEC &HE54C**でOKです．具体的な例を**リスト4**に示します．$01AC～$01AFをすべて255にすると，割り込み間隔がなんと2年8箇月になります（気が狂う）．

なお，この一見便利なINTERVAL TIMERは悪質ないたずらをします．一度INTERVAL TIMERを設定すると，プログラムが終ってもSUB・CPUはMAINCPUへFIRQを送り続けます．

MAIN CPUがカセット入出力のシリアル・パラレル変換をやっているときに，FIRQがかかるとほとんどの場合，力強くERRORに飛びます．カセットの入出力を行なう前には必ず"INTERVAL OFF"を実行しましょう．

PART 5
その他

F-BASICの**LOAD？**は，メモリ上の内容とカセットテープの内容の比較ではなくて，カセット上のチェック・サムのチェックなんですね．つまり，メモリ上に何もなくても**LOAD？**を行なえます．

FM-8の**BEEP**（一定時間BELを鳴らす）はハードでやっているのでしょうか？ **POKE &HFD03, &H61**で**BEEP**が一定時間鳴ります．また**BEEP**が鳴っている間は，次の**BEEP**を受けつけませんので，**BEEP**を何回か連続して実行しても1度しか鳴りません．

現在自作の逆アセンブラでF-BASICインタープリタを解読中です．いままでに分ったことを以下に記します．

＊主なデータ領域

$8B68～$8BB4	RESET時に表示されるタイトル.
～$8BE7	"COM 0", "BUB 0" アドレス・テーブル
$8BFD～$8C47	RESET時のファンクション・キーの内容
$8C48～$8CA2	ワーク・エリアへ転送するデータ
$8CDA～	ASCIIコード. "Memory Parity Error"
$8CF0～$8D69	関数の処理ルーチンのアドレス・テーブル
$8D6A～$8FB3	ASCIIコード. キーワード・テーブル
$8FB4～$904B	ステートメントの処理ルーチンのアドレス・テーブル
$904C～$91FC	中間言語—キーワードの対応表
$91FD～$9230	キーワードの最初の文字—キーワード・テーブルの対応表
$9231～$9242	ASCIIコード "Ready", "Break"
$A804～$AACC	ASCIIコード エラーメッセージ
$BFB8～$BFDC	ASCIIコード RANDOMIZEのプロンプト・メッセージ
$CA23～$CA65	省略形キーワード・テーブル
$CEED～$CF09	ASCIIコード "Searching", "Found :", " Skip:"
$F304～$F33B	BIOS内部の各処理ルーチンのアドレス・テーブル

＊使えそうなサブルーチン

$9FDF	1行出力	Xreg＋1のアドレスから0があるまで文字列として出力
$A01A	1行出力	XregからBregの文字数だけ文字列として出力
$D352	1文字出力	Aregの値をASCIIコードとする文字を出力
$9F97	CRLF	改行する
$AFE9	HEX1出力	Aregの値を16進文字列で出力
$AFE3	HEX2出力	Dregの値を16進文字列で出力
$B96D	10進出力	Dregの値を10進文字列で出力
$DD69	INKEY	押されたキーのASCIIコードをAregに読みこむ
$DA50	1行入力	CRまでにキー入力された文字列がASCIIコードで$043Dから入る. Xregは$043Cになる.

PART 6

おわりに

どうやら富士通は，FM-8にはバブルかディスクがつながっているものとしてシステムを作ったようです．カセットだけではどうも不自由な面が多いですね．ディップ・スイッチを切り換えてROMを殺しても，ブートROMまで切り換わってしまうために，プログラムを走らせることは非常に困難です．

ブートROMの1/4は予備として未使用となっているのは，どういう理由でしょう．将来はマイクロ・ディスクのブート用となるのかもしれませんが，どうせ未使用なら暫定的にカセットからのブートを考えてもよかったと思います．

現在私はカセットからブートするためのROMを焼こうか，ハードを改造しようかと考えています．

＊I/Oポート

$FD00 bit 0	カセットへの出力	
bit 1	MOTOR用リレーのスイッチ出力	
bit 2	プリンタへのストローブ出力	
$FD01	プリンタへのデータ出力	
$FD02 bit 0	プリンタのBUSY信号入力	
$FD03 bit 6	BEEP 出力	BEEP=POKE &HFD03, 65 BEEP 1 =POKE &HFD03, 129
bit 7	BEEP ON/OFF	BEEP 0 =POKE & HFD03, 1
$FD04 bit 1	STOP KEY 入力	
$FD05 bit 7	入力：SUB CPU BUSY 出力：SUB CPU HALT	
$FD08～$FD0B	音声合成用	
$FD0C	A—D変換器への入出力	
$FD10～$FD17	バブルカセット入出力	
$FD18～$FD1F	フロッピーディスク入出力	
$FD20～$FD21	漢字ROMのアドレス出力(漢字コードではありません)	
$FD22～$FD23	漢字データ入力（漢字の横方向16ドット）	

＊その他

$890A	BASIC コールド・スタート．フリーエリアをすべてクリアします．
$D8B8	BASIC ホット・スタート（STOP—RESET）
$F2D8	BIOS ENTRY
$92AB	ERROR Bregにエラーコードを入れてジャンプ
$AFA2	MONITOR ENTRY. 機械語プログラムの最後はこのアドレスに飛ばすと便利です.

＊割り込み

$01D7	SWI	F-BASICでは使用していません． 通常$3B：RTIが書き込まれています．
$01D4	SWI 2	
$01D1	SWI 3	
$01E0→$CC57	FIRQ	STOP, INTERVALなどの割り込み.
$01D0→$D5AB	IRQ	サブCPUからの割りこみ.
$01DA→$NMI	NMI	Memory Parity Errorが出ます.

なお，簡単なバージョン・チェックです．私のFM-8は3283676でした．

FOR I=&H 8000 TO &H FBFF : A#=A#+PEEK(I):NEXT

ROM→RAM BASICプログラム・リスト 1

```
                                NAM    ROMRAM
7F00                            ORG    $7F00
7F00 86    3B          ENTRY    LDA    #$3B
7F02 B7    01E0                 STA    $01E0
7F05 8E    0800                 LDX    #$0800
7F08 E6    89 8000     LOOP1    LDB    $8000,X
7F0C E7    80                   STB    X+
7F0E 8C    7C00                 CMPX   #$7C00
7F11 26    F5    7F08           BNE    LOOP1
7F13 86    81                   LDA    #$81
7F15 B7    FD03                 STA    $FD03
7F18 B6    FD04        LOOP2    LDA    $FD04
7F1B 85    02                   BITA   #$02
7F1D 26    F9    7F18           BNE    LOOP2
7F1F 8E    0800                 LDX    #$0800
7F22 4F                         CLRA
7F23 E6    84          LOOP3    LDB    ,X
7F25 E7    89 8000              STB    $8000,X
7F29 A7    80                   STA    X+
7F2B 8C    7C00                 CMPX   #$7C00
7F2E 26    F3    7F23           BNE    LOOP3
7F30 8E    8800                 LDX    #$8800
7F33 7E    892D                 JMP    $892D
                                END
```

REMARKプログラム・リスト 2

```
                                NAM    REMARK
0500                            ORG    $500
0500 FE    0033                 LDU    >$0033
0503 8E    3A8D                 LDX    #$3A8D
0506 CC    4040                 LDD    #$4040
0509 108E  0D11                 LDY    #$0D11
050D AC    44          LOOP     CMPX   4,U
050F 26    08    0519           BNE    SKIP
0511 10A3  46                   CMPD   6,U
0514 26    03    0519           BNE    SKIP
0516 10AF  46                   STY    6,U
0519 EE    C4          SKIP     LDU    ,U
051B 26    F0    050D           BNE    LOOP
051D 39                         RTS
                                END
```

図1

```
<REMARK, REMARK> マニュアルで行なうときの手順
A) 操作したいプログラム (例)
   ⟆
   120 GOTO 5000
   130 '@@  1Byte Peek  ←色をつけたい行
   140 D=PEEK (A) : A=A+1
B) 色をつけたい行のみのLISTをとる.
   LIST 130
   130 '@@ 1Byte Peek
   Ready
C) MONITORにし, $00A2と$00A3番地の内容を見る
   MON
   *MA2
   00A2 09— } 130行は$0946番地より
   00A3 46— } 格納されていることが分る
D) Mコマンドで$0946番地からの内容を書き変える
   *M946
   0946  09— } 次の行が格納されているアドレス
   0947  5A— } リンク・ポインタ
   0948  00— } 行番号
   0949  82— }   $0082=130
   094A  3A— } "," の中間言語
   094B  8D— } "REM" だと$8Cの1バイトになる
   094C  40— [11] } $11=DC1
   094D  40— [04] } カラーコード. $04ならば緑色になる.
                ← この部分を書き変える
   094E  20— }
   094F  31— } 以後はREMARK文中の文字列が
   0950  42— }   ASCIIコードで入っている.
E) BASICへ戻りLISTをとる
   ⟆
   120  GOTO 5000              ← この部分のみ緑色になる
   130 '[1Byte Peek]
   140  D=PEEK(A) : A=A+1
   ⟆
```

LET'S GO MUSICプログラム・リスト3

```
10 '4 **** FM-8 MUSIC PROGRAM ****
20 CLEAR 300,&H2FFF:A=&H3000
30 READ A$
          *1          *1
40 IF A$<>"@" THEN POKE A,VAL("&H"+A$):A=A+1:GOTO 30
50 RESTORE 500
60 READ I
70 IF I<256 THEN POKE A,I:A=A+1:GOTO60
80 POKE A,0:POKE A+1,0
90 PRINT"DATA SET READY ":A$=INPUT$(1)
100 ON INTERVAL GOSUB 120:INTERVAL 1:INTERVAL ON
110 EXEC &H3000
120 END
             *1
130 '4 ** PROGRAM DATA **
140 DATA 30,8C,14
150 DATA BF,FF,F6
160 DATA 30,8C,42
170 DATA CE,FD,00
180 DATA 20,0B
190 DATA CC,01,E0
200 DATA FD,FF,F6
210 DATA 7E,94,83
220 DATA 35,24
230 DATA F6,FD,04
240 DATA C5,02
250 DATA 27,EE
260 DATA 30,02
270 DATA A6,01
280 DATA 27,E8
290 DATA B7,01,AF
300 DATA 34,70
310 DATA BD,E5,4C
320 DATA 35,70
330 DATA 1C,BF
340 DATA A6,84
350 DATA 27,FC
360 DATA 1A,40
370 DATA 4A
380 DATA 26,FD
390 DATA 86,C3
400 DATA A7,C4
410 DATA A6,84
420 DATA 4A
430 DATA 26,FD
440 DATA 86,C2
450 DATA A7,C4
460 DATA 1C,BF
470 DATA 20,E6
480 DATA 10,00
490 DATA @
500 '1 ***  URUSEIYATSURA  ***
510 '1 *** RAM NO LOVESONG ***
520 '6 * ANMARISOWASOWA...
530 DATA 8,136
540 DATA 8,144
550 DATA 8,136
560 DATA 8,114
570 DATA 8,121
580 DATA 8,136
590 DATA 8,153
600 DATA 8,172
610 DATA 8,183
620 DATA 8,153
630 DATA 8,114
640 DATA 32,136
650 DATA 8,0
660 '6 * ANATAWAITSUDEMO...
670 DATA 8,172
680 DATA 8,183
690 DATA 8,172
700 DATA 8,136
710 DATA 8,153
720 DATA 8,172
730 DATA 8,183
740 DATA 8,205
750 DATA 8,183
760 DATA 8,153
770 DATA 8,114
780 DATA 32,136
790 DATA 8,0
800 '6 * YOSOMIOSURUNOWA.
810 DATA 8,136
820 DATA 8,144
830 DATA 8,136
840 DATA 8,114
850 DATA 8,121
860 DATA 8,136
870 DATA 8,153
880 DATA 8,172
890 DATA 8,183
900 DATA 8,153
910 DATA 8,114
920 DATA 32,136
930 DATA 8,0
940 '6 * WATASHIGAICHIBAN...
950 DATA 8,136
960 DATA 8,144
970 DATA 8,136
980 DATA 8,114
990 DATA 8,121
1000 DATA 8,136
1010 DATA 8,121
1020 DATA 8,153
1030 DATA 8,114
1040 DATA 8,114
1050 DATA 8,121
1060 DATA 24,114
1070 DATA 16,0
1080 '6 * SUKIYO SUKIYO...
1090 DATA 6,114
1100 DATA 2,0
1110 DATA 6,114
1120 DATA 2,0
1130 DATA 8,114
1140 DATA 8,0
1150 DATA 6,55
1160 DATA 2,0
1170 DATA 6,55
1180 DATA 2,0
1190 DATA 8,55
1200 DATA 8,0
1210 DATA 6,136
1220 DATA 2,0
1230 DATA 6,136
1240 DATA 2,0
1250 DATA 8,136
1260 DATA 8,0
1270 DATA 6,66
1280 DATA 2,0
1290 DATA 6,66
1300 DATA 2,0
1310 DATA 8,66
1320 DATA 8,0
1330 DATA 6,114
1340 DATA 2,0
1350 DATA 6,114
1360 DATA 2,0
1370 DATA 8,114
1380 DATA 8,0
1390 DATA 4,55
1400 DATA 4,58
1410 DATA 4,55
1420 DATA 4,58
1430 DATA 8,55
1440 DATA 8,0
1450 DATA 6,136
1460 DATA 2,0
1470 DATA 6,136
1480 DATA 2,0
1490 DATA 8,136
1500 DATA 8,0
1510 DATA 4,66
1520 DATA 4,70
1530 DATA 4,66
1540 DATA 4,70
1550 DATA 8,66
1560 DATA 8,0
1570 '6 * SUKIYO SUKIYO.
1580 DATA 6,114
1590 DATA 2,0
1600 DATA 6,114
1610 DATA 2,0
```

リスト3

```
1620 DATA 8,114
1630 DATA 8,0
1640 DATA 6,55
1650 DATA 2,0
1660 DATA 6,55
1670 DATA 2,0
1680 DATA 8,55
1690 DATA 8,0
1700 DATA 6,136
1710 DATA 2,0
1720 DATA 6,136
1730 DATA 2,0
1740 DATA 8,136
1750 DATA 8,0
1760 DATA 6,66
1770 DATA 2,0
1780 DATA 6,66
1790 DATA 2.0
1800 DATA 8,66
1810 DATA 8,0
1820 DATA 6,114
1830 DATA 2,0
1840 DATA 6,114
1850 DATA 2,0
1860 DATA 8,114
1870 DATA 8,0
1880 DATA 4,55
1890 DATA 4,58
1900 DATA 4,55
1910 DATA 4,58
1920 DATA 8,55
1930 DATA 8,0
1940 DATA 6,136
1950 DATA 2,0
1960 DATA 6,136
1970 DATA 2,0
1980 DATA 8,136
1990 DATA 8,0
2000 DATA 4,66
2010 DATA 4,70
2020 DATA 4,66
2030 DATA 4,70
2040 DATA 8,66
2050 DATA 8,0
2060 '6 * HOSHITACHIGA...
2070 DATA 12,121
2080 DATA 4,0
2090 DATA 12,121
2100 DATA 4,0
2110 DATA 12,121
2120 DATA 4,0
2130 DATA 8,121
2140 DATA 8,121
2150 DATA 8,0
2160 DATA 8,121
2170 DATA 8,0
2180 DATA 8,121
2190 DATA 16,114
2200 DATA 14,121
2210 DATA 2,0
2220 DATA 12,136
2230 DATA 4,0
2240 DATA 8,136
2250 DATA 8,153
2260 DATA 8,0
2270 DATA 8,89
2280 DATA 8,84
2290 DATA 8,75
2300 DATA 8,66
2310 DATA 8,70
2320 DATA 8,66
2330 DATA 8,75
2340 DATA 8,0
2350 DATA 8,89
2360 DATA 8,84
2370 DATA 8,75
2380 '6 * YUMEMIRUNO,,,
2390 DATA 12,121
2400 DATA 4,0
2410 DATA 12,121
2420 DATA 4,0
2430 DATA 12,121
2440 DATA 4,0
2450 DATA 8.121
2460 DATA 8,121
2470 DATA 8,0
2480 DATA 8,121
2490 DATA 8,0
2500 DATA 8,121
2510 DATA 16,114
2520 DATA 14,121
2530 DATA 2,0
2540 DATA 12,136
2550 DATA 4,0
2560 DATA 8,136
2570 DATA 8,153
2580 DATA 8,0
2590 DATA 8,89
2600 DATA 8,84
2610 DATA 8,75
2620 DATA 8,66
2630 DATA 8,70
2640 DATA 8,66
2650 DATA 8,75
2660 DATA 8,0
2670 DATA 8,89
2680 DATA 8,84
2690 DATA 8,75
2700 '6 * AISHITEMO...
2710 DATA 12,136
2720 DATA 4,0
2730 DATA 12,136
2740 DATA 4,0
2750 DATA 12,136
2760 DATA 4,0
2770 DATA 8,136
2780 DATA 8,136
2790 DATA 8,0
2800 DATA 8,136
2810 DATA 8,0
2820 DATA 8,136
2830 DATA 16,128
2840 DATA 14,136
2850 DATA 2,0
2860 DATA 11,153
2870 DATA 11,172
2880 DATA 10,153
2890 DATA 16,136
2900 DATA 8,172
2910 DATA 16,205
2920 DATA 56,0
2930 '6 * IMAGOROWA...
2940 DATA 12,121
2950 DATA 4,0
2960 DATA 12,121
2970 DATA 4,0
2980 DATA 12,121
2990 DATA 4,0
3000 DATA 8,121
3010 DATA 8,121
3020 DATA 8,0
3030 DATA 8,121
3040 DATA 8,0
3050 DATA 8,121
3060 DATA 16,114
3070 DATA 8,121
3080 DATA 8,121
3090 DATA 64,114
3100 DATA 24,121
3110 DATA 8.0
3120 '6 * AA OTOKONO...
3130 DATA 32,153
3140 DATA 64,89
3150 DATA 8,0
3160 DATA 8,89
3170 DATA 8,95
3180 DATA 8,89
3190 DATA 16,84
3200 DATA 16,89
3210 DATA 8,89
3220 DATA 8,95
3230 DATA 8,89
3240 DATA 80,101
3250 DATA 8,0
3260 '6 * IKUTSUMO...
3270 DATA 16,136
3280 DATA 16,101
3290 DATA 8,101
3300 DATA 40,101
3310 DATA 8,0
3320 DATA 8,101
3330 DATA 8,107
3340 DATA 8,101
3350 DATA 8,89
3360 DATA 8,89
3370 DATA 16,101
3380 DATA 8,114
3390 DATA 8,114
3400 DATA 8,101
3410 DATA 80,89
3420 DATA 8,0
3430 '6 * AA ACHIKOCHINI...
3440 DATA 16,153
3450 DATA 64,89
3460 DATA 8,0
3470 DATA 8,89
3480 DATA 8,95
3490 DATA 8,89
3500 DATA 16,84
3510 DATA 16,89
3520 DATA 16,89
3530 DATA 8,89
3540 DATA 24,101
3550 DATA 16,114
3560 DATA 40,101
3570 DATA 8,0
3580 '6 * WATASHIO...
3590 DATA 16,136
3600 DATA 16,101
3610 DATA 8,101
3620 DATA 40,101
3630 DATA 8,0
3640 DATA 8,101
3650 DATA 8,107
3660 DATA 8,101
3670 DATA 8,89
3680 DATA 24,101
3690 DATA 64,114
3700 DATA 8,55
3710 DATA 8,49
3720 DATA 8,43
3730 DATA 8,40
3740 DATA 8,49
3750 DATA 8,43
3760 DATA 8,40
3770 DATA 8.35
3780 '6 * ANMARISOWASOWA...
3790 DATA 8,136
3800 DATA 8,144
3810 DATA 8,136
3820 DATA 8,114
3830 DATA 8,121
3840 DATA 8,136
3850 DATA 8,153
3860 DATA 8,172
3870 DATA 8,183
3880 DATA 8,153
3890 DATA 8,114
3900 DATA 32,136
3910 DATA 8,0
3920 '6 * ANATAWAITSUDEMO...
3930 DATA 8,172
3940 DATA 8,183
3950 DATA 8,172
3960 DATA 8,136
3970 DATA 8,153
3980 DATA 8,172
3990 DATA 8,183
4000 DATA 8,205
4010 DATA 8,183
4020 DATA 8,153
4030 DATA 8,114
4040 DATA 32,136
4050 DATA 8,0
4060 '6 * YOSOMIOSURUNOWA...
4070 DATA 8,136
4080 DATA 8,144
4090 DATA 8,136
4100 DATA 8,114
4110 DATA 8,121
4120 DATA 8,136
4130 DATA 8,153
4140 DATA 8,172
4150 DATA 8,183
4160 DATA 8,153
```

リスト3

```
4170 DATA 8,114
4180 DATA 32,136
4190 DATA 8,0
4200 '6 * WATASHIGAICHIBAN...
4210 DATA 8,136
4220 DATA 8,144
4230 DATA 8,136
4240 DATA 8,114
4250 DATA 8,121
4260 DATA 8,136
4270 DATA 8,121
4280 DATA 8,153
4290 DATA 8,114
4300 DATA 8,114
4310 DATA 8,121
4320 DATA 24,114
4330 DATA 16,0
4340 '6 * ICHIBAN SUKIYO...
4350 DATA 8,114
4360 DATA 8,114
4370 DATA 8,121
4380 DATA 8,114
4390 DATA 8,0
4400 DATA 8,121
4410 DATA 8,0
4420 DATA 8,121
4430 DATA 48,114
4440 DATA 16,0
4450 DATA 256,0
```

表1 LET'S GO MUSIC音階データ

	C^2	C^3	C^4	C^5
C	231	114	55	26
C#	218	107	52	24
D	205	101	49	22
D#	194	95	46	21
E	183	89	43	19
F	172	84	40	18
F#	162	79	38	17
G	153	75	35	16
G#	144	70	33	15
A	136	66	31	14
A#	128	62	29	13
B	121	58	27	12

出力される周波数＝CLOCK/{2＊(19＋5＊(DATA))}　(Hz)
CLOCK＝4915200 Hz

リスト4

```
10 ON INTERVAL GOSUB100  } INTERVAL設定
20 INTERVAL ON           }
30 POKE &H01AF, 25       } 割り込み間隔を0.5秒にセット
40 EXEC &HE54C           }   25×20msec＝0.5sec
50 GOTO50
100 PRINT"@";            } 割り込みルーチン
110 RETURN               } 0.5秒おきに"@"を表示
```

図2　実行結果

```
8DE2 — AUTO            =9D       ADDRESS : A3F3
↑         └キーワード    └中間言語コード        ↑
└キーワードの文字列が      そのキーワードが実行された┘
  入っているアドレス        ときに飛ぶアドレス
```

図2を出力するプログラム

```
100 FORA=0TO50STEP2:B=PEEK(&H91FD+A)*256+PEEK(&H91FE+A)
110 WHILEPEEK(B)>0:C=PEEK(B)*256+PEEK(B+1):PRINTHEX$(C);" - ";
120 WHILEPEEK(C)<128:PRINTCHR$(PEEK(C));:C=C+1:WEND:PRINTCHR$(PEEK(C)AND127);TAB
(16);" = ";
130 D=PEEK(B+2):IFD<128THEND=D+128:PRINT"FF ";:E=&H8CF0 ELSEIFD<&HCC THENE=&H8FB
4 ELSEE=0
140 PRINTRIGHT$("0"+HEX$(D),2);:IFE>0THEND=(127ANDD)*2+E:PRINTTAB(25);"ADDRESS :
"RIGHT$("000"+HEX$(PEEK(D)*256+PEEK(D+1)),4)ELSEPRINT
150 B=B+3:WEND:PRINT
160 NEXT
```

FM-8 音声入出力プログラム

Voice Monitor



■笠作貴弥

I/O '81年5月号にMIMICシステムが紹介されてから，MZ-80バージョンが色々と発表されていますが，FM-8でも特別な改造，ハードの追加なしで，音声入出力ができることがわかりました．

プログラムは，最も基本的なもので，単に指定アドレス間に音声データをストアしたり，指定アドレス間の音声データを出力するだけのものになっています．だから具体的応用は，みなさんにおまかせします．

使用方法

まず，プログラムを入力します．そして，図1のように，オーディオ・カセット・インターフェイス・ケーブルの赤いプラグをアンプに接続し，白いプラグをいつも使っているカセットレコーダーのイヤホン端子に，黒いプラグは，外して置きます．そうしたら，プログラムをRUNさせます．そうすると，コマンド待ちになるので，コマンド表に従って，色々と試してください．再生される音声にきっと感動すると思います．

プログラムについて

プログラムは，BASIC部分については，ほとんど説明の必要はないと思います．マシン語サブルーチンの方を少し説明すると，このルーチンは，I/O 5月号のp.175のリストをほとんどそのまま6809用に書き換えただけのものです．$E50，$E51に上位，下位の順でスタート・アドレス，$E52，$E53に入出力するバイト数をセットして，$E00をコールするとその間のデータを出力します．また同様に$E2Bをコールすると，指定アドレス間にデータを取り込みます．

それからリスト2のように380行のデータ文を書き換えると，内蔵スピーカから音声を出力することが，可能になりますが，かなり音質が低下します．

最後に

今回のものは本当に基本的なものなので，現在Ver.2.0として，I/O11月号のようにストリング変数中のデータを発声するプログラムを開発中で，今音声データ（可愛いい女の子の声です）を編集中です．乞ご期待！

◘参考文献

1) I/O, '81年5月号 "MIMICシステム"

2) F-BASIC文法書

図1

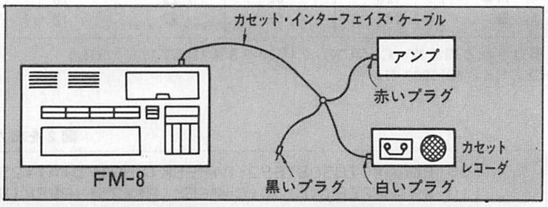


表1 コマンド

INPUT：指定アドレス間にカセットレコーダーからの音をストアします．

OUTPUT：指定アドレス間のデータをアンプに出力します．

TEMPO：再生時のクロック・レートを設定します．1～255を入力してください．ノーマルは，9です．

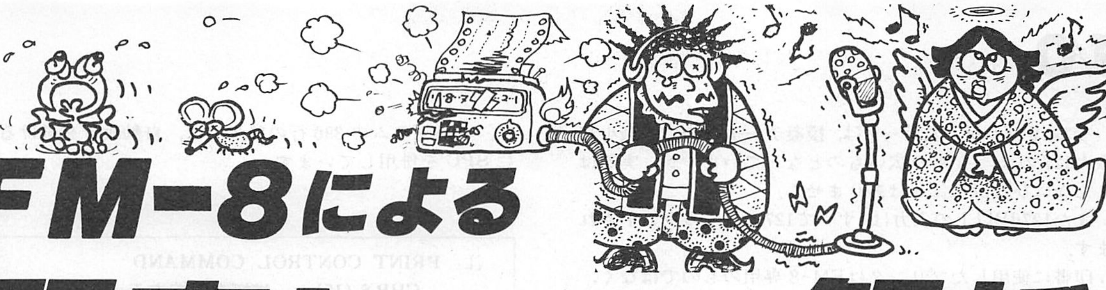
コマンド待ちのとき，単に RETURN を押すと，プログラムが終了します．

リスト1

```
100 '**********************************
110 '*                                *
120 '*  VOICE Monitor Ver.1.0         *
130 '*                                *
140 '* 1981 Nov.8th Written BY JI1ETF *
150 '*                                *
160 '**********************************
170 '
180 CLEAR 50,&HDFF
190 FOR I=&HE00 TO &HE4F:READ A$:POKE I,VAL("&H"+A$):NEXT
200 WIDTH 40,20:CONSOLE 0,20,0,1
210 PRINT" *** VOICE Monitor Ver.1.0 ***":PRINT
220 PRINT" FREE AREA $F00-$7EFF"
230 LINEINPUT":";A$
240 IF A$="" THEN END
250 IF A$="TEMPO" THEN 280
260 IF A$="INPUT" THEN 290
270 IF A$="OUTPUT" THEN 300 ELSE PRINT CHR$(7):GOTO 230
280 INPUT A:POKE &HE1B,A:GOTO 230
290 GOSUB 310:EXEC &HE2B:EXEC &HE00:BEEP 0:GOTO 230
300 GOSUB 310:EXEC &HE00:BEEP 0:GOTO 230
310 INPUT"Start Address ";A$:S=VAL("&H"+A$)
320 IF S<&HF00 OR S>&H7EFF THEN PRINT CHR$(7):GOTO 310
330 POKE &HE50,S/256:POKE &HE51,S MOD 256
340 INPUT"End Address   ";A$:E=VAL("&H"+A$)
350 IF S>E OR E<&HF00 OR E>&H7EFF THEN PRINT CHR$(7):GOTO
    340
360 E=E-S+1:POKE &HE52,E/256:POKE &HE53,E MOD 256:RETURN
370 DATA 34,36,BE,0E,50,10,BE,0E,52,A6,80,C6,08,48,34,02
380 DATA 25,04,86,01,20,01,4F,B7,FD,00,86,09,4A,26,FD,35
390 DATA 02,5A,26,E9,31,3F,26,E1,35,36,39,34,36,BE,0E,50
400 DATA 10,BE,0E,52,78,FD,02,25,FB,C6,08,78,FD,02,69,84
410 DATA 86,0C,4A,26,FD,5A,26,F3,30,01,31,3F,26,EB,20,D8
```

リスト2

```
380 DATA 25,04,86,81,20,01,4F,B7,FD,03,86,09,4A,26,FD,35
```



FM-8による「騒音レベル」解析

●玉村卓也

FM-8には他機種にみられないユニークな機能がいくつもありますが，なかでもタイマーによるインタラプトとか，A/D変換とかが，BASICレベルで組み込まれているのは大変魅力的です．A/D変換はアナログ・ポートとして，5PINのDINプラグが背面に用意されていて，その接続は，図1のようになっています．

5Vの電源が供給され，4チャネルの入力が可能ということは，2個のX-Yジョイスティック，あるいは，4個のパドル入力ができるわけで，ゲーム狂さんには垂涎のまとだと思います．

たまたま，これらの機能にピッタリ，フィットする仕事があったので，さっそくトライしてみました．

アナログ・ポート(ANalogue PORT)の読み込みは，1バイトの値（0～255）の関数として，

ANPORT (*c*, *l*)

の形でとりいれられます．*c*はチャネル0～3で*l*は入力レベル0あるいは1です．

l=0のとき	0～0.625V RMS
l=1のとき	0～2.5V RMS

一方，インターバル・タイマは，秒単位で1秒～65,535秒(≒18.2時間)の設定ができますが，長時間の方はTIMEの設定による割り込みも可能なので，むしろ1/50秒～1300秒のような使い方ができる方が，より望ましいように思います．ここでは最短の1秒間隔で計測し，最大計測数を200としたので，3分20秒となりますが，途中での打切りも可能としています．

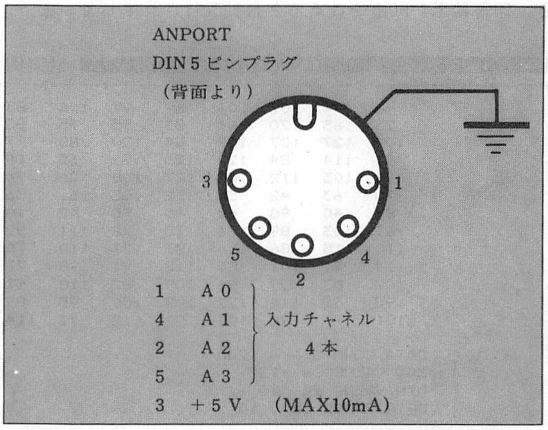
計測には騒音計の出力を直接入れる方法，一度テープに録音しておいて再生出力を入れる方法，これらをペン・レコーダを通し，そのペン出力を入れる方法などいろいろありますが，まずそれらの出力レベルを確認してください．

このプログラムでは，高感度の方，すなわち0～5/8V(0.625V）のレンジをえらんでありますが，NAGRA KUDELSKIあたりの計測用テレコになると，出力は0.1V程度なので，20dBぐらいのプリアンプが必要です．プリアンプ，アッテネータは特性のよいものを選んでください．1秒間隔の計測ですから，一度レコーディングしておいて，何度か再生して平均をとるのがよいと思われます．

RUNすると

INPUTが0となるSignalのレベルを聞いてきますが，ペン・レコーダ出力などで，実際の出力が0となる前に足切りをして出力を0にしているような機器から，データを入力する場合は，その機器で，何dBから足切りしているかを予かじめ調べておいてください．

図1



テレコなどリニヤーに出力されるものであったら，ここは0を回答します．

なお，プログラムの120行では，標準回答として，0を表示させてこれを入力に使うつもりだったのですが，成功しませんでした．

入力レベルのCALIBRATEのために，テープの頭には，114dBの連続信号を1～2分入れておきます．

これが114になっているのは，たまたま私の使った騒音計のCALレベルが114dB（または10dB刻みでこれに＋－）であったためで，これに外のCALレベルの機器を常用されるときは，修正してください．

10回計測して平均が114dBであれば，テストを通過させていますが，このテスト方法はもう一工夫した方がよかったと思います．

次いでスペース・バーを押した時点から計測が始まり，読み取り値がCRT上に表示されます．200個読み取ると自動的に終りますが，途中でESCキーを押せば，その時点までのデータで計算過程にすすみます．

計算結果は，例示のように標本読み取り値，標本数，最大値，最小値，巾およびL5，L10，L50，L90，L95の値を印書しさらに，度数分布図を画きます．

FM-8にはこの上，HARDC命令で画面をそっくりプリンタに出力できる機能もあるので，累積度数曲線をLINE命令で，CRT上に画き，それをハード・コピーさせるようなおもしろい応用が可能です．

L5値とは，この累積度数が，標本数の5％を越えたときの入力の値のことで，以下同様です．

なお，例示に使ったデータは，模擬データなので，最小10，最大127 のような幅の広いものとなっていますが，実際はこんなにひろがることはありません．

また127dB以上の入力は，すべて127としてカウントされます．

印書に使用したプリンタはFM-8 専用のものではなく，BASIC MASTER L3用のＴＰ80Ｔを流用しているので，これ以外の機種でハード・コピーをとるときは，プリンタに対する制御コマンドをそれらに合うものに修正してください．

駄足ながら，度数分布図は，縮少モードで，タテ，ヨコとも圧縮しています．縮少モードでは，ヨコは132 桁と見なされます．しかしBASICの方のTAB命令では，現在プリンタが縮小モードで働いているかどうかを知る手段は仕組まれていません．

したがって，TABによって横位置をコントロールさせるとき，印書結果が80桁をこえると自動改行されてしまいます．プログラムの 390 行の後半では，自動改行を避けるためにSPC を併用しています．

注　PRINT CONTROL COMMAND

CHR＄(15)　　横幅を圧縮する

CHR＄(18)　　横幅を正常に戻す

CHR＄(27)；"A"；CHR＄(n)

タテ方向の紙送り幅をｎに設定する．ただしｎは，1インチ（通常の６行分）につき72だから正常値はｎ＝12で，文字自体の幅は６に相当するためｎ＝６とすれば，印字の上下はすきまができない．

CHR＄(31)　　グラフィック文字印書モード

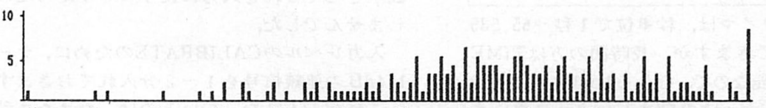
（これはＢ・Ｍ Ｌ３のノンインターレース・モードに相当する）

計測例１，２

```
  90   94  107  103   69   97  104   83   90   73   73   66   50   38   46   57
  23   65   70   84   83   95   75   82   97  100   83  104   97   81   96  120
 101  127  127  117   84  120   88   77  117   80   90   79   60  127   75   64
  81  114   84  124  125   72   65   69  114   93   75   96  120   87   49  110
  94  103  112  103  127  100   85   74   55   79   77   78   73   85   80   50
  70   63   42   55   72  105   61   56   21   98   57   52   71   59   67   86
  89   40   90   62   45   90   42   10  101  105   45   88   81   59   96   89
  69  103   89   76   73   78  111   92   79   84   93  108  121  116  120   87
 110  115   86  114  127  127  110   78  103  101  124   91   81   67   97   76
 114   81  101   83  112   91   98   77   88   94   94   96   92   85  107   79
 107   83   89   86  109   82  118   91   76   99  103   85   93  119  110  127
  92   32  116  127  109  100   95   89  110  105   87   91   77   86  111   81
 101   77   98  102   97  112   73  114

DATA COUNTS Are  200
MAX VALUE is  127
MIN VALUE is  10
DATA RANGE is  118

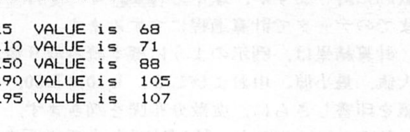
L5   VALUE is  46
L10  VALUE is  59
L50  VALUE is  89
L90  VALUE is  117
L95  VALUE is  124
```




```
 103  117   99   91   87   96   84   99   86   91   98   93   94   86   87   90
  88   94   92   88   81   84   86   80   82   87   83   81   87   74   70   69
  72   75   72   72   74   69   80   71   74   78   86   92  103   94   91   90
  92  103  108  105  105  110  107   98   97  104  105  107  106  107  107  108
 101  105  101  110  104   97   92   91   94   88   98   98   83   76   75  105
  77   77   68   64   63   61   58   59   82   69   83   76   91   98   98   86
  81   75   78   82   81   81   79   90   90

DATA COUNTS Are  105
MAX VALUE is  117
MIN VALUE is  58
DATA RANGE is  60

L5   VALUE is  68
L10  VALUE is  71
L50  VALUE is  88
L90  VALUE is  105
L95  VALUE is  107
```



騒音レベル解析プログラム

```
10 GOTO50: REM SOUND LEVEL ANALIZER
20 A=INT(ANPORT(0,0)*R+Z):PRINT USING"#####";A;:IF A>MAX THEN MAX=A
30 IF A<MIN THEN MIN=A
40 D(N)=A:N=N+1:RETURN
50 CLS:DIM D(200),L$(4),L(5):N=0:MAX=0:MIN=9999
60 L$(0)="L5 ":L$(1)="L10":L$(2)="L50":L$(3)="L90":L$(4)="L95"
70 ON INTERVAL GOSUB20:INTERVAL 1
110 OPEN"O",#2,"LPTO:":PRINT#2,CHR$(31);
120 PRINT"INPUTｶﾞ Oﾄﾅﾙ Signalﾉ ﾚﾍﾞﾙ ﾊ  "O;:LOCATEPOS(0)-5,CSRLIN:INPUT Z:R=(127.
5-Z)/255
130 PRINT"ANALOGUE PORTﾉ CHANNEL O ﾆ 114dbﾉ Standard Signalｦ ｲﾚﾃｸﾀﾞｻｲ"
150 PRINT"ｿﾉ SIGNALﾉ ﾖﾐﾄﾘ ｶﾞ 114ﾆ ﾅﾙﾖｳﾆ Volumeｦ ﾁｮｳｾﾂ ｼﾃｸﾀﾞｻｲ"
170 A=0:FOR I=1TO10:A=A+ANPORT(0,0):GOSUB490:NEXT:LOCATE5,8,1:A=INT(A*R/10+Z):PR
INTA:IFA<>114THEN170
180 PRINT"SPACE Barｦ ｵｽﾄ ｿｸﾃｲｦ ｶｲｼ ｼﾏｽ"
190 PRINT"ｿｸﾃｲ ｼﾞｶﾝﾊ 200"CHR$(&HF7)"ﾃﾞｽ":PRINT"ﾄﾁｭｳﾃﾞ ｳﾁｷﾘﾀｲﾄｷﾊ ESC ｷｰｦ ｵｼﾃｸﾀﾞｻｲ
"
200 K$=INPUT$(1):IFK$<>" "THEN200
210 N=0:INTERVAL ON
220 IF INKEY$=CHR$(27) THEN 230 ELSE IF N<200THEN220
230 INTERVAL OFF:PRINT:K=MAX-MIN:FORI=0TON-1:PRINT#2,USING "#####";D(I);:NEXT:PR
INT#2:PRINT#2
240 PRINT#2,"DATA COUNTS Are "N
250 PRINT#2,"MAX VALUE is "MAX
260 PRINT#2,"MIN VALUE is "MIN
270 PRINT#2,"DATA RANGE is "K+1
280 DIMS(K):L(0)=N*.05:L(1)=N*.1:L(2)=N*.5:L(3)=N*.9:L(4)=N*.95:L(5)=N
300 FOR I=0TON-1:J=D(I)-MIN:S(J)=S(J)+1:NEXT
310 S=0:H=0:PRINT#2:MAX=0
320 FOR J=0TOK:IFS(J)>MAX THEN MAX=S(J)
330 S=S+S(J):IFS>L(H)THEN PRINT#2,L$(H)" VALUE is "J+MIN:H=H+1
340 NEXT:PRINT#2:PRINT#2:PRINT#2,CHR$(15);CHR$(27);"A";CHR$(6):T=(INT(MAX/5)+1)*
5:FOR T=T TO1 STEP-1
350 IF INT(T/5)*5=T THEN PRINT#2,USING "#####";T;:PRINT#2," ┤"; ELSE PRINT#2,TAB
(6)"|";
360 FORJ=0 TO K:IF S(J)>=T THEN PRINT#2,"▮";ELSE PRINT#2," ";
370 NEXT:PRINT#2:NEXT
380 PRINT#2,TAB(6);:FORJ=0TOK+1:PRINT#2,CHR$(&H94);:NEXT:PRINT#2
390 PRINT#2,TAB(6);:FORJ=0TOK+1:M=J+MIN:IF INT(M/10)*10=M THENIFJ<69THENPRINT#2,
TAB(J+6);M;ELSE IF M<110 THENPRINT#2,SPC(6);M;ELSEPRINT#2,SPC(5);M;
400 NEXT:PRINT#2
410 PRINT#2,CHR$(18):PRINT#2,CHR$(27);"A";CHR$(12):PRINT#2:CLOSE2:END
490 FORJ=0TO250:NEXT:RETURN:TIMER
```

付録FM-8メモリ・ダンプ・プログラム

```
10 CLEAR3000:OPEN"O",#2,"LPTO:"
20 PRINT#2,CHR$(27);"N";CHR$(2);
30 INPUT"START ADDRESS,END ADDRESS";B$,E$
40 B=VAL("&H"+B$):E=VAL("&H"+E$)
50 PRINT#2,RIGHT$("000"+HEX$(B),4);"  ";
60 B$="":C=0
70 FOR I=B TO B+7:A=PEEK(I):IF A>=32 AND A<127 THEN B$=B$+CHR$(A)ELSEB$=B$+"."
80 PRINT#2,RIGHT$("0"+HEX$(A),2);" ";:C=C+A:NEXT:PRINT#2," ";:B$=B$+" "
90 FOR I=B+8 TO B+15:A=PEEK(I):IF A>=32AND A<127 THEN B$=B$+CHR$(A)ELSEB$=B$+"."
100 PRINT#2,RIGHT$("0"+HEX$(A),2);" ";:C=C+A:NEXT
110 PRINT#2,"  ";RIGHT$("0"+HEX$(C AND 255),2);"  ";B$
120 B=B+16:IF B>E THENPRINT#2:PRINT#2:PRINT#2:GOTO 30 ELSE 50
```

FM-8

N-BASIC → F-BASIC TEXT CONVERTER



■前橋高校物理部　TQJ

ＦＭ-8が発売されて半年になりますが，まだソフトの方はあまり出まわっておりません．そこで，プログラムの多いPC-8001のものを利用できるように，このTEXT-CONVERTERを作りました．

特徴として次のような事柄があげられます．

❶ N-BASICのテープがそのまま読める(機械語のテープにSAVEし直す必要はない)．

❷ グラフィック命令については，画面に合わせて自動的に変更される（これにより，コンバート後の手作業がかなり軽減される）．

❸ コンバート後，すぐＲＵＮできる状態になる．

コンバータの作り方

リスト1を入れてください．ただし，999行から1900行までと，10,000行以後の〝Ａ＝&Ｈ～〟と〝；～〟は入れる必要はありません(入れても影響はありませんが…)．

ＲＵＮさせると，チェック・サムを出力してきます．ＤＡＴＡ文最後のサムと一致するか確認してください．最後にＳＡＶＥをするかどうか聞いてくるので，必要があれば Y キーを押してください．コンバータのテープができあがります．

使い方

本プログラムは，7800Ｈから7ＦＦＦＨを使用します．よって，ＲＥＳＥＴの後，ＣＬＥＡＲで領域を確保します．そして，コンバータのテープを読み込みます．

次にＰＣのテープをセットして，ＥＸＥＣ32000 RET (LOAD RET のかわり）を入力します．

ファイル名はいりません．一番最初のものをＬＯＡＤします．ヘッダーを見つけると，

```
Found：〔ファイル・ネーム〕
```

と表示して，プログラム本体を読み始めます．完了すると，

```
Ready
```

が出ます．

あとは「修正表」に従って，手直しをします．不適合な所はエラーが出るので，ＲＵＮさせながらやってもいいと思います．

変更点

1 まず，テープのスピードが合わないことが予想されます．ＰＣのように，ハードでＰＬＬ追尾はできないので，781FHと78F8Hの値を調節します（テープが速いときには小さくする）．

2 このコンバータでは，ＦＭ-8に合うように，下記のとおり自動変更を行ないます．

★COLOR　第２，第３パラメータを削除…………〔a〕
★CONSOLE　第3，第4パラメータを各々0に……〔b〕
★PSET / PRESET / LINE　Ｘ成分４倍，Ｙ成分２倍（LINEはグラフィックのみ）…………………〔c〕

以上のものが不要のときは次のように変更します．

〔ａ〕が不要のとき…7F1BH　26Ｈ→20Ｈ
〔ｂ〕が不要のとき…7F5FH　26Ｈ→20Ｈ
〔ｃ〕が不要のとき…7F82H　21Ｈ→7ＥＨ

3 読み込みにエラーばかりするときは，

7855Ｈ	25Ｈ→21Ｈ
7862Ｈ 7866Ｈ 786AＨ	24Ｈ→21Ｈ

というようにすると，読めるときがあります．

エラーについて

異常があるときは以下のようなエラーメッセージが出ます．

★Syntax Error
N-BASICにおいて文法がおかしい．

★Undefind Line No.
ＩＤ変換の際，行番号がみつからない．16ＫＰＣと32ＫＰＣを混用したときにまれに起きる．

★Out Of Memory
メモリが不足した．

★Bad File Mode
テープがN-BASICのものでない．ヘッダーの読み取りミス．

★Device I/O Error
テープの読み取り不良．

注★印の２項以外のエラーはやり直してもたぶん無駄でし

よう．しかし，まれにSyntax Errorのときに，RESET をして始めからやり直すと，うまくいくことがあります．

◘参考文献

1）PC-8001 リファレンス・カード
2）MICRO-8 文法書
3）I/O誌,
　　FM-8 エントリ・マップ
　　FM-8 特集記事
4）FM-8 中間言語表，マイコン，'81年10月号
5）N-BASIC入門，PC-8001 中間言語表

修正表

ＰＣのリスト	コンバート後	修　　正
PEEK	SCREEN	画面以外に使われているものはダメ．X成分，Y成分にわけて書き直す．
POKE	｜POKE	画面なら，LOCATE＋PRINTで！
USR	｜USR	まずできない．マシン語の研究を．
OUT	｜POKE	POKEでI/Oアドレスへ．
INP	STOP	KEYのサーチなら，INKEY$で代用．
COLOR,,1	COLOR	色指定以外のCOLOR文は削除．
WIDTH 36 or 72	そのまま	36→40，72→80
LINE 5, 6	そのまま	線を引く以外のLINE文は削除．
PUT @ GET @	そのまま	Gオプション付きなら，X４倍，Y２倍をする．ただし，配列の大きさに注意！
ERASE	｜E	プログラム次第ではRUN．
DATA文の後ろのマルチ・ステートメント	破壊	削除，または，ＰＣ側で入れ換えておく．
DISK関係	STOP	まったく考慮に入れていない．

テキストコンバータ プログラム・リスト

```
1 '   *********************************
2 '   *                               *
3 '   *      N-Basic => F-Basic       *
4 '   *         TEXT CONVERTER        *
5 '   *              Ver. 4.3         *
6 '   *                               *
7 '   *            82/01/15           *
8 '   *                               *
9 '   *********************************
100   CLEAR,&H7800
110   FOR I=&H7800 TO &H7FFF STEP 16
120     S=0
130     FOR J=I TO I+15
140       READ A$
150       A=VAL("&H"+A$)
160       POKE J , A
165       'IF A<>PEEK(J) THEN STOP
170       S=S+A
180     NEXT
190     PRINTRIGHT$("0"+HEX$(S),2)+"  ";
200   NEXT
300   PRINT"SAVE? (Y/N)"
400   IF INPUT$(1)="Y" THEN SAVEM"PC.CON43",&H7800,&H7FFF,32000
500   END
999   WIDTH80 : KEY 10,CHR$(127)+CHR$(31)+CHR$(127)+CHR$(31)
1000  INPUT"Add=",A$ : I=VAL("&H"+A$) : INPUT"Line",L
1100  FOR I=I TO I+16*15 STEP 16
1200    A$=STR$(L)+"   A=&H"+HEX$(I)+" : DATA  "
1250    S=0 : L=L+10
1300    FOR J=I TO I+14
1400      A$=A$+RIGHT$("0"+HEX$(PEEK(J)),2)+","
1500      S=S+PEEK(J)
1600    NEXT
1700    A$=A$+RIGHT$("0"+HEX$(PEEK(J)),2)+" ; "+RIGHT$("0"+HEX$(S+PEEK(J)),2)
1800    PRINTA$
1900  NEXT
9998  '
9999  '   Add               +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F : Sum
10000  A=&H7800 : DATA  7E,7D,00,BD,78,88,30,36,9F,35,BD,D9,BB,7E,7D,00 ; 3E
10010  A=&H7810 : DATA  96,02,2B,FC,8E,FF,FF,30,01,96,02,2A,FA,8C,00,18 ; DC
10020  A=&H7820 : DATA  39,46,6F,75,6E,64,3A,20,47,61,6C,61,78,69,00,41 ; 26
10030  A=&H7830 : DATA  54,34,1F,86,FD,1F,8B,8E,08,00,17,00,A5,24,0A,30 ; 84
10040  A=&H7840 : DATA  1F,26,F7,BD,CB,25,FC,35,DF,8D,C5,25,33,C6,08,8D ; CE
10050  A=&H7850 : DATA  BF,25,0D,BD,BB,25,29,1C,FE,66,A4,5A,26,F1,20,15 ; 51
10060  A=&H7860 : DATA  8D,AE,24,1C,8D,AA,24,18,8D,A6,24,14,1A,01,66,A4 ; 7E
10070  A=&H7870 : DATA  5A,26,DC,20,00,8D,99,25,00,35,9F,20,B4,7E,92,A0 ; 1F
10080  A=&H7880 : DATA  35,1F,7E,CD,A1,7E,D1,B8,7E,78,9D,30,8C,96,A6,80 ; 52
10090  A=&H7890 : DATA  A7,A0,10,8C,04,00,25,F6,8E,00,FF,00,00,8E,03,C0 ; E0
10100  A=&H78A0 : DATA  BD,40,24,F9,30,1F,26,F8,10,8E,78,28,8E,78,20,86 ; 41
10110  A=&H78B0 : DATA  D3,C6,0A,8D,C6,A1,A4,26,CC,5A,26,F7,C6,06,8D,BB ; B8
10120  A=&H78C0 : DATA  31,21,5A,26,F9,6F,A4,BD,9F,DF,10,9E,33,0F,35,5F ; 9D
10130  A=&H78D0 : DATA  8D,A9,A6,A0,10,9C,45,24,A4,4D,26,F3,5C,C1,03,20 ; DB
10140  A=&H78E0 : DATA  EE,39,B6,FD,02,84,80,26,F9,CE,FF,FF,33,41,B6,FD ; F2
10150  A=&H78F0 : DATA  02,84,80,27,F7,11,83,00,18,39,00,00,00,7E,95,29 ; 45
10155  '
10156  '   Add               +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F : Sum
10160  A=&H7900 : DATA  DE,45,A6,B2,A7,C2,9C,33,26,F8,1F,31,EC,84,27,5E ; E6
10170  A=&H7910 : DATA  30,04,8D,34,9C,45,24,70,4D,27,F1,81,0D,26,F3,EC ; 62
10180  A=&H7920 : DATA  1E,9F,37,1F,31,10,A3,81,27,12,10,AE,1E,27,56,30 ; 3A
10190  A=&H7930 : DATA  02,DD,39,8D,13,4D,26,FB,DC,39,20,E9,EC,84,9E,37 ; 89
10200  A=&H7940 : DATA  ED,1E,86,0E,A7,1D,20,CA,A6,80,81,0D,27,32,81,0E ; E9
```

テキストコンバータ・プログラム・リスト

```
10210    A=&H7950 : DATA  27,2E,81,0F,27,2C,81,1C,27,26,81,1D,27,20,81,1F ; A7
10220    A=&H7960 : DATA  27,1A,81,22,26,09,A6,80,27,04,81,22,26,F8,39,81 ; DF
10230    A=&H7970 : DATA  E4,27,04,81,8F,26,F7,A6,80,26,FC,39,30,04,30,02 ; 23
10240    A=&H7980 : DATA  30,01,30,01,39,7E,95,29,7E,97,40,00,00,00,00,00 ; 2C
10250    A=&H7990 : DATA  00,00,00,00,00,00,00,00,BD,7F,3F,C1,28,26,F9 ; 83
10260    A=&H79A0 : DATA  E7,80,BD,7F,3F,C1,2C,27,04,8D,39,20,F5,30,1F,CC ; F0
10270    A=&H79B0 : DATA  29,DB,ED,81,CC,FE,01,ED,81,CC,04,2C,ED,81,C6,28 ; 03
10280    A=&H79C0 : DATA  E7,80,BD,7F,3F,C1,2C,27,08,C1,29,27,04,8D,15,20 ; D5
10290    A=&H79D0 : DATA  F1,30,1F,CC,29,DB,ED,81,CC,FE,01,ED,81,C6,02,E7 ; 66
10300    A=&H79E0 : DATA  80,33,5F,39,C1,28,26,13,0F,00,0C,00,BD,7F,3F,C1 ; C4
10310    A=&H79F0 : DATA  28,27,F7,C1,29,26,F5,0A,00,26,F1,39,00,00,00,00 ; A5
10315    '
10316    '   Add                +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F : Sum
10320    A=&H7A00 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 ; 00
10330    A=&H7A10 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 ; 00
10340    A=&H7A20 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 ; 00
10350    A=&H7A30 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 ; 00
10360    A=&H7A40 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 ; 00
10370    A=&H7A50 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 ; 00
10380    A=&H7A60 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 ; 00
10390    A=&H7A70 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 ; 00
10400    A=&H7A80 : DATA  00,9C,9D,9E,80,81,82,85,A4,89,86,87,88,8A,8B,83 ; 39
10410    A=&H7A90 : DATA  00,84,8D,8E,8F,90,91,A0,96,98,97,00,00,00,00,92 ; 46
10420    A=&H7AA0 : DATA  93,94,95,00,00,00,00,00,00,99,00,00,00,00,00,00 ; 55
10430    A=&H7AB0 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 ; 00
10440    A=&H7AC0 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 ; 00
10450    A=&H7AD0 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 ; 00
10460    A=&H7AE0 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 ; 00
10470    A=&H7AF0 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 ; 00
10475    '
10476    '   Add                +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F : Sum
10480    A=&H7B00 : DATA  00,00,00,01,00,02,00,03,00,04,00,05,00,06,00,07 ; 1C
10490    A=&H7B10 : DATA  00,08,00,09,00,0A,FE,02,FE,02,00,8E,FE,F2,FE,01 ; 98
10500    A=&H7B20 : DATA  00,10,FE,30,FE,31,FE,32,FE,33,FE,34,FE,35,FE,36 ; 67
10510    A=&H7B30 : DATA  FE,37,FE,38,FE,39,FE,3A,FE,02,FE,04,FE,08,FE,08 ; E8
10520    A=&H7B40 : DATA  00,20,00,21,FE,22,00,23,00,24,00,25,00,26,00,27 ; 1A
10530    A=&H7B50 : DATA  00,28,00,29,00,2A,00,2B,00,2C,00,2D,00,2E,00,2F ; 5C
10540    A=&H7B60 : DATA  00,30,00,31,00,32,00,33,00,34,00,35,00,36,00,37 ; 9C
10550    A=&H7B70 : DATA  00,38,00,39,00,3A,00,3B,00,3C,00,3D,00,3E,00,3F ; DC
10560    A=&H7B80 : DATA  00,40,00,41,00,42,00,43,00,44,FE,45,00,46,00,47 ; 1A
10570    A=&H7B90 : DATA  00,48,00,49,00,4A,00,4B,00,4C,00,4D,00,4E,00,4F ; 5C
10580    A=&H7BA0 : DATA  00,50,00,51,00,52,00,53,00,54,00,55,00,56,00,57 ; 9C
10590    A=&H7BB0 : DATA  00,58,00,59,00,5A,00,5B,00,5C,00,5D,00,5E,00,5F ; DC
10600    A=&H7BC0 : DATA  00,60,00,61,00,62,00,63,00,64,00,65,00,66,00,67 ; 1C
10610    A=&H7BD0 : DATA  00,68,00,69,00,6A,00,6B,00,6C,00,6D,00,6E,00,6F ; 5C
10620    A=&H7BE0 : DATA  00,70,00,71,00,72,00,73,00,74,00,75,00,76,00,77 ; 9C
10630    A=&H7BF0 : DATA  00,78,00,79,00,7A,00,7B,00,7C,00,7D,00,7E,00,7F ; DC
10635    '
10636    '   Add                +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F : Sum
10640    A=&H7C00 : DATA  00,8E,00,80,00,81,00,82,FE,83,FF,A6,00,84,00,85 ; 40
10650    A=&H7C10 : DATA  00,86,87,CD,00,88,00,89,00,8A,87,CE,00,8B,FE,8C ; DF
10660    A=&H7C20 : DATA  7D,8E,00,B9,00,BC,00,BB,00,C0,00,97,00,8E,00,B7 ; D7
10670    A=&H7C30 : DATA  00,B8,00,BA,00,AA,00,AB,7B,B8,7D,B9,7D,BB,00,A5 ; 0D
10680    A=&H7C40 : DATA  00,A0,00,8F,00,90,00,91,00,92,7D,45,00,9B,00,9C ; DB
10690    A=&H7C50 : DATA  00,9E,00,9D,00,99,00,93,00,94,00,95,00,96,00,B6 ; DC
10700    A=&H7C60 : DATA  00,A7,00,A6,00,B4,00,8E,00,B2,00,B5,00,9F,00,A2 ; 37
10710    A=&H7C70 : DATA  00,8E,00,A8,00,8E,00,8E,00,8E,00,8E,00,8E,00,8E ; 8A
10720    A=&H7C80 : DATA  00,8E,00,8E,00,8E,00,8E,00,8E,7D,AE,00,8E,00,C1 ; 40
10730    A=&H7C90 : DATA  00,C2,00,8E,00,AF,8E,AB,00,AC,00,B0,00,8E,00,CA ; EC
10740    A=&H7CA0 : DATA  00,8E,FF,8E,8E,AA,00,8E,00,8E,00,A3,00,8E,20,CD ; 8D
10750    A=&H7CB0 : DATA  20,D6,00,CC,20,D8,7D,D2,00,CF,00,D0,20,D7,00,D3 ; 72
10760    A=&H7CC0 : DATA  00,D4,FF,A3,00,D1,FF,9F,FE,8D,FF,A2,FF,A7,00,8E ; 45
10770    A=&H7CD0 : DATA  00,8E,FF,A5,01,54,01,44,00,8E,00,8E,00,8E,FF,A8 ; 1D
10780    A=&H7CE0 : DATA  00,E5,00,E6,00,E7,00,D9,00,DA,00,DB,00,DC,00,DD ; F9
10790    A=&H7CF0 : DATA  20,DE,20,DF,20,E0,20,E1,20,E2,20,E3,00,E4,00,8E ; 75
10795    '
10796    '   Add                +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F : Sum
10800    A=&H7D00 : DATA  10,CE,FC,7F,BD,D9,B8,86,FF,97,35,BD,78,9D,0D,35 ; 0C
10810    A=&H7D10 : DATA  26,F9,30,36,9F,35,BD,D9,BB,9E,35,BD,79,00,7E,7F ; B0
10820    A=&H7D20 : DATA  00,ED,81,BD,CA,66,30,02,9F,35,BD,CA,F1,7E,94,89 ; 74
10830    A=&H7D30 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 ; 00
10840    A=&H7D40 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 ; 00
10850    A=&H7D50 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 ; 00
10860    A=&H7D60 : DATA  00,00,00,00,7E,7D,00,12,00,00,00,00,00,00,00,00 ; 0D
10870    A=&H7D70 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 ; 00
10880    A=&H7D80 : DATA  81,01,26,15,C6,FF,E7,80,C1,54,26,05,CC,AA,24,20 ; E3
10890    A=&H7D90 : DATA  05,CC,A9,24,12,12,12,12,12,7E,7E,82,00,00,00,00 ; 76
10900    A=&H7DA0 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 ; 00
10910    A=&H7DB0 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 ; 00
10920    A=&H7DC0 : DATA  C1,22,26,19,E7,80,E6,C0,27,07,C1,22,26,F6,7E,7E ; 58
10930    A=&H7DD0 : DATA  B1,7E,7E,A1,00,00,12,00,00,00,00,00,00,C1,83,27 ; CB
10940    A=&H7DE0 : DATA  08,C1,8C,27,04,C1,8D,26,0E,E7,80,A6,C0,A7,80,26 ; 1C
10950    A=&H7DF0 : DATA  FA,7E,7E,8F,7E,7E,92,ED,81,C1,01,27,0D,C1,04,27 ; 63
10955    '
10956    '   Add                +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F : Sum
10960    A=&H7E00 : DATA  0E,C1,08,27,17,20,2E,7E,7E,80,E6,C0,7E,7E,84,EC ; F1
```

```
10970   A=&H7E10 : DATA  C1,34,06,EC,C1,1E,89,ED,81,35,06,20,EA,EC,46,1E ; 52
10980   A=&H7E20 : DATA  89,ED,81,EC,44,1E,89,ED,81,EC,42,1E,89,ED,81,EC ; 6B
10990   A=&H7E30 : DATA  C4,33,48,20,D2,C1,02,27,09,C4,0F,30,1F,86,01,7E ; 4B
11000   A=&H7E40 : DATA  7E,82,EC,C1,20,ED,00,00,00,00,00,00,00,00,00,00 ; BA
11010   A=&H7E50 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 ; 00
11020   A=&H7E60 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 ; 00
11030   A=&H7E70 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,ED,81,EC,C1 ; 1B
11040   A=&H7E80 : DATA  1E,89,A7,80,E7,80,1C,FD,1A,01,39,12,1C,FC,39,1A ; 1F
11050   A=&H7E90 : DATA  02,39,DF,35,9C,35,12,12,25,03,7E,92,A0,E6,C0,26 ; E8
11060   A=&H7EA0 : DATA  04,6F,80,20,EA,C1,FF,27,3F,4F,58,49,EC,AB,4D,26 ; 1D
11070   A=&H7EB0 : DATA  04,E7,80,20,D7,81,FE,26,17,C1,45,26,0C,A6,C4,81 ; 41
11080   A=&H7EC0 : DATA  A1,26,EE,33,41,C6,8F,20,16,C1,F2,27,AF,7E,7D,C0 ; F8
11090   A=&H7ED0 : DATA  81,20,27,03,7E,7D,80,A6,1F,81,20,27,D4,20,15,86 ; 62
11100   A=&H7EE0 : DATA  20,A7,80,86,3A,20,9B,12,E6,C0,86,FF,E6,AB,26,92 ; 48
11110   A=&H7EF0 : DATA  C6,8E,20,BD,86,20,7E,7E,82,00,00,00,00,7E,7D,21 ; 71
11115   '
11116   '   Add             +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F : Sum
11120   A=&H7F00 : DATA  10,8E,7B,00,9E,33,EC,C1,27,F3,ED,81,EC,C1,1E,89 ; 73
11130   A=&H7F10 : DATA  ED,81,BD,7E,92,29,EF,25,F9,C1,B5,26,40,BD,7F,3F ; C8
11140   A=&H7F20 : DATA  C1,2C,26,F9,9F,37,BD,7E,92,29,0D,25,F9,C1,3A,26 ; 24
11150   A=&H7F30 : DATA  F5,9E,37,E7,1F,7E,7F,12,9E,37,6F,1F,7E,7F,06,BD ; 02
11160   A=&H7F40 : DATA  7E,92,29,12,25,F9,C1,20,27,F5,C1,3A,27,01,39,10 ; D2
11170   A=&H7F50 : DATA  CE,FC,7D,7E,7F,12,10,CE,FC,7D,7E,7F,06,C1,A5,26 ; 3D
11180   A=&H7F60 : DATA  21,BD,7F,3F,C1,2C,26,F9,BD,7F,3F,C1,2C,26,F9,CC ; FB
11190   A=&H7F70 : DATA  FE,01,ED,81,CC,00,2C,ED,81,CC,FE,01,ED,81,6F,81 ; FC
11200   A=&H7F80 : DATA  20,A2,21,7F,12,C1,A6,27,04,C1,A7,26,05,BD,79,99 ; 68
11210   A=&H7F90 : DATA  20,80,C1,B6,26,FA,9F,37,DF,39,BD,7F,3F,BD,79,E4 ; BA
11220   A=&H7FA0 : DATA  C1,29,26,F6,BD,7F,3F,BD,79,E4,C1,29,26,F6,BD,7F ; DD
11230   A=&H7FB0 : DATA  3F,C1,2C,26,F9,BD,7F,3F,C4,80,27,11,9E,37,DE,39 ; 2E
11240   A=&H7FC0 : DATA  BD,79,99,BD,79,99,BD,7F,3F,C1,2C,26,F9,BD,7F,3F ; A0
11250   A=&H7FD0 : DATA  C1,2C,26,F9,E7,80,7E,7F,12,00,00,00,00,00,00,00 ; 82
11260   A=&H7FE0 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 ; 00
11270   A=&H7FF0 : DATA  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,12 ; 12
```

補　足

『N-BASIC➡F-BASICテキスト・コンバータ』の改良

N➡F-BASICコンバータはNewバージョンのテープを読み込まないようなので訂正したものです．以下にその変更点を掲げます．

（田崎　明久）

改良点

アドレス	旧データ	新データ
7845	25	20
7846	FC	F3
78DE	03	0A
78DF	20	25
78E0	EE	EG
7D13	36	3A

FM-8

AUTO-2001（カセット版）

■COMPAC

MZB用のプログラムレス言語“AUTO-2001”の移植版です．MZBとFM-8では，BASICフリーエリアに違いがあるので，今回はカセット版とします．もちろん，入出力先を換えるとバブル版にもなります．ディスクBASICのもとでは，フリーエリアが少なく実用的でありません．

小型のデータの処理に有効です．さらに大型データに対応するには，工夫が必要です．全部BASICなので，解読して自分用の簡易言語処理システムを作ると便利でしょう．

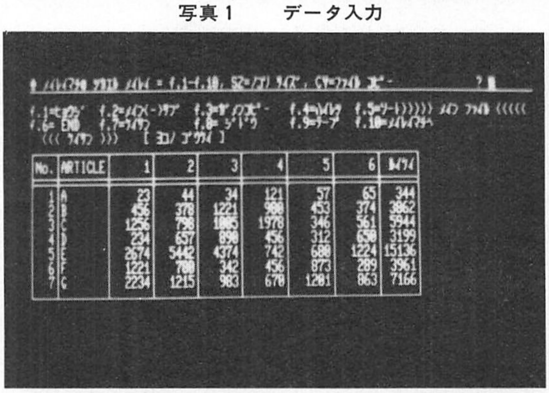
特長

通常のものは画面の制約から74桁程度しか処理できませんが，実用面からはプリンタの能力一ぱいの136桁まで処理できるようにしました．そのため，独立したファイルを2つ用意して，あたかも1つのファイルのように自由に呼び出し，データ加工，プリントができるように工夫しています．

もう一つはAUTO-2001の名に恥じないように自動機能でデータ入力・計算・ソート・配列替え・印刷が自動的に行なえるようにした点で，一度手順を登録すれば全部自分でやってくれます．

プログラム構成

写真1　データ入力



下表の機能を持った2つのプログラムで構成されています．できれば1本にしたかったのですが，メモリ制約上データ・エリアが実用的でなくなるので2本に分けました．

CMT-1はファイル作成用とサーチ，グラフ，インサツと一応の機能をそなえているので，通常の作表業務には対応できます．このCMT-1で作ったデータ・ファイルをフルに活用するのが，CMT-3で，自動機能と大ファイル機能（136桁までのコントロール）を持ち，通常の事務処理はすべて自動化できます．

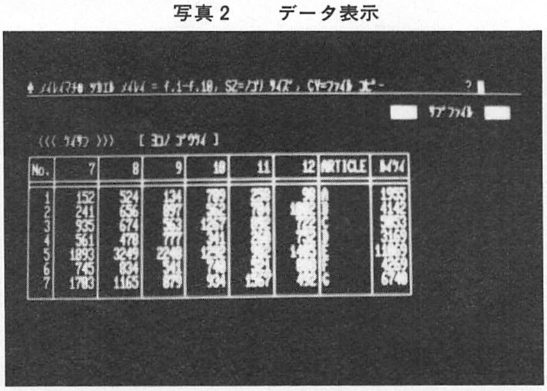
命令	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	SZ	CL	CY
機能	表示	メイン・サブ切替	印刷・ガメンコピー	編集・配列替	ソート サーチ	グラフ	計算	自動	テープ（I/O）	メイレイマチへ	ノコリサイズ	クリア	ファイルコピー
CMT-1	○	○	印刷	編集	サーチ	○	—	—	○	○	○	○	—
CMT-3	○	○	コピー	配列替	ソート	—	○	○	○	○	○	—	○

使用法

細かい説明はすべて画面上に表示されます．

主な機能はプログラマブル・ファンクション・キーの1～10を押すことで動作します．このファンクション・キー

写真2　データ表示



の機能の説明は，命令入力待ちのとき[CR]を押すと消去され，再度[CR]を押すと表示されます．以下の説明で[F1]～[F10]は，このファンクション・キーのことです．

最初に，[F10]を押すといつでも命令入力待ちに復帰できることを憶えておいてください(エラーでBASICに抜けたときも)．

項目は，メイン・ファイルであればABC…，サブファイルであればabc…の記号で指定します．

CMT-1の使用法

CMT-1は，データ作成と印刷・グラフ・サーチ機能を備えたもので，英文ワード・プロセッサ，通常の作表はこのプログラムだけでOKです．

❶まず，プログラムをRUNすると新規データ・ファイル作成か前のデータ・ファイルを使うのかを聞いてきます．ここでは新規作成の[1]としましょう．

❷日付・ファイル名・コメントをキーインすると，これがこのファイルの個有のデータになります．以後のテープ(I/O)はこのファイル名で行われます．

❸項目表設定モードではファイルの横方向の項目を設定しますが，74桁，26項目まで設定できます．残りサイズ表示に注意しながら項目名と桁数を設定してください．設定の終りは"]"をキーインします．そこで設定OKかを聞いてきます．

❹項目設定を完了するには[3]をキーインするとデータNo.1の入力待ちになります．データは255行まで入れられます．

❺入力モードでは"へ"で上と同じ"："で左と同じデータを入れることができます．

"<"[CR]でその行の最初から">"[CR]でエデット・モードになり，上の行のデータの修正ができます．

空欄は","でできます．入力終了は"["をキーインします．

❻これで命令待ちになりますが，この段階で作成されたファイルはメイン・ファイルに入っています．

サブファイルに別のデータ・ファイルを作る場合は，まず，[F2]を押してサブファイルにコントロールを移します．[F2]は押すたびに切り換わり，そのときのファイル表示が右上に出ます．

❼コントロールをサブファイルにした状態でCL[RET]をキーインすると，

サブファイル・クリア…[1]，オール・クリア…[2]

と聞いてきます．メイン・ファイルのデータはそのままにしておきます．

[1]をキーインすると新規作成か，前のファイルを使うかと聞いてくるので，[1]をキーインすればまったく同様にしてもう一つのファイルをサブファイル上に作ることができます．

❽命令待ちにおける各命令は右上に表示しているファイルに対して行なわれるので，注意してください．

❾[F1]はファイル内容の表示です．[F1]をキーインすると，表示する行番号を聞いてくるので，

イ）全部表示する場合は"－"か"・"をキーイン．
ロ）10行までは　"－10"
ハ）20行以降は　"20－"
ニ）10行から20行は　"10－20"
ホ）15行のみは　"15"

とキーインします．この行番号指定方法は他のたとえばサクジョやグラフの場合も使います．

❿表示の途中任意のキーを押せばストップ・スタートできます．ストップさせているとき"へ"をキーインすると，エディット・モードになり，表示しているデータの修正ができます．

⓫この[F1]と[F2]を交互に使えば，両ファイルの内容を表示するので，あたかも1つのファイルのように使えます(実際に1つのファイルとして自由にデータ加工できるわけですが，それはCMT-3で行ないます)．

⓬次に，[F3]をキーインしてみます．

ガメンコピー－[1]　アテナインサツ－[2]　データインサツ－[3]　ブンショインサツ－[4]

と聞いてきます．[1]をキーインすれば画面コピーを行ない，[2]をキーインすればアテナインサツになります．[3]のデータインサツと[4]の文書インサツでは，ファイル名・日付の印刷と自動ページ替えが行なわれます．

⓭[F4]はデータ・ファイルを編集するためのもので，

テイセイ－[1]　アペンド－[2]　データ　ツイカ，サクジョ－[3]

となっています．[1]をキーインすればエディット・モードになり，データだけでなく項目名も修正できます．[2]はファイルへのアペンドで，コントロール下のファイルへ他のファイル・データをアペンドしたり，テープにある新しいデータをアペンドしたりできます．[3]はデータの追加削除で任意のデータを削除したり，任意の行番号の位置にデータを挿入したりできます．

⓮[F5]はサーチ(データ・ケンサク)で，

モジレツまたはスウジレツ…[1]　ジョウケンケンサク…[2]と聞いてきます．[1]の場合はファイルの全データの中にある文字列または数字列を検索します．[2]をキーイン

写真3　自動計算式入力

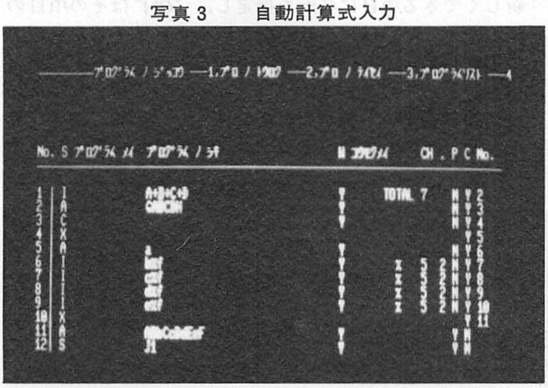
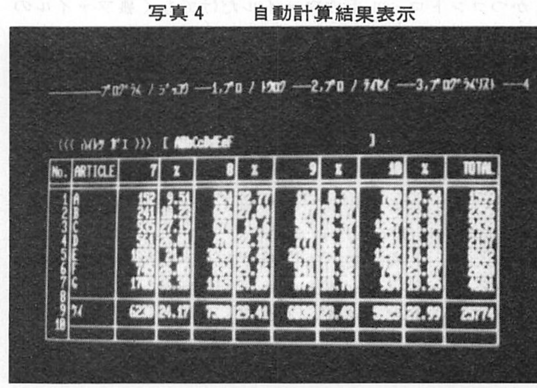


写真4　自動計算結果表示



するとジョウケンハ？と聞いてきます．同時に項目表を表示するので，その項目表の下に書いてあるＡＢＣで，たとえば“A＝100”または“C＞＝25”と次々にキーインすると，キーインされた条件のAND検索を行ないます．

⓯ F6 はグラフ作成で項目表を表示するとともにグラフにする項目，グラフにする範囲，グラフのフルスケールを順にキーインすると第１項目を見出しにしたグラフを表示します．

グラフ表示中任意のキーでストップ・スタートができストップ中または表示終了時に F3 をキーインすると画面コピーします．表示終了後任意のキーを押すと，インサツするかと聞いてきます． Y とキーインするとプリンタに印刷します．スケールは自動スケーリングでマイナスのデータも表示します．

⓰ F9 はテープ（I/O）で，

テープにセーブ－1　アラタニテープヨリロード－2と出ます． 1 は現在コントロール下にあるファイルをテープにセーブしますが，ファイル名，コメント，日付を自動的にセーブできます． 2 は現在コントロール下にあるファイルを消して新らしくデータをロードしますが，不用意に貴重なデータを消さないように確認してきます．

なお，ロードするファイル名が分らない場合は“@”をファイル名としてキーインすると最初にあるテープ・データをロードします．

⓱ＳＺ RET をキーインするとコントロール下のファイルにデータをあと何行追加できるかを示してくれます．

CMT-3の使用法

CMT-3はCMT-1で作成されたデータ・ファイルを元にいろいろな処理を行なうパッケージで，AUTO-2001の特長はこれで発揮できます．命令表で示してある機能のうちCMT-1と同じものは省略します．

❶プログラムをロードすると，初めにファイルメイを聞いてくるので，テープをセットしてファイルメイをキーインしてください．ファイルメイを忘れた場合は“@”をキーインすると最初に見つけたファイルをロードします．

❷ F3 は画面コピーを行ないます．

❸ F4 は項目の配列替および追加（同時にできます）で現在の項目表が出るので，その項目を必要な順に並べ替えることができます．また，ＡＢＣ＋とすると項目の追加になります．項目データは同じものを入れる，または個別に入れることができます．

この機能はAUTOの特長で74桁を超えた範囲まで可能で，かつコントロール下のファイルだけでなく裏ファイルの項目も自由に扱えます．我々はこれを『大ファイル・コントロール』と名付けていますが，この機能によって，見かけ上52項目の148桁までのファイル・コントロールが可能です．

つまり，表示されている表の記号を任意に選んで，たとえばAaBdefCとキーインすれば，その項目を配列替えしてくれます．この場合74桁を超えると表示しませんが，印刷は136桁まで行ないます．

❹ F5 はソート機能で，項目表に表示された項目を大きい順か小さい順に並べ替えます．この場合，ファイルは書き替えられるので注意してください．

ソートは数字だけでなく文字列に対しても行なわれるので，ＡＢＣ，イロハ順への並べ替えもＯＫです．ただし，マルチソート機能はありません．

❺ F7 は計算機能で，

項目計算－1　タテの合計－2　ヨコの合計－3

となっています． 1 を押すと項目表が表示され，演算内容を聞いてきます．演算結果はコントロール下のファイルに項目として追加されます．演算は大ファイルに対して行なえるので，フルにファイルを活用してください．

使える演算子は“＋”，“－”，“＊”，“／”，“％”，“へ”，です．％はパーセント，へは巾（べき）計算です．また，数字も扱えるし，負数も扱えます．たとえば，－0.8＊Ｂ＋ＡもＯＫです．

小数以下何桁を出すか聞いて来るので，たとえば 2 とキーインすると小数以下３桁を４捨５入して表示します（これは計算の全ステップで行なわれるので要注意）．

“／”，“％”の計算の場合，分母が０のときは計算結果を０にするので，注意してください．また，“へ”を不用意に使うとすぐ桁オーバーエラーになります．

❻ F8 は自動処理を行なう部分で，この機能でユーザーの工夫次第でほとんどの業務がキー１つですべて自動的に行なわれます．

プログラムの実行－1　プログラムの登録－2　プログラムの修正－3　プログラム・リスト表示－4

がありますが， F8 をキーインすると自動的にプログラム・リストを表示してきます．このリストは任意のキーを押すことにより，ストップ・スタートします． 1 をキーインするとスタートさせるプログラムNo.を聞いてくるので，それをキーインすると，プログラムの指示に従って次々に計算・ソート・配置換え・印刷を行なってくれます．

❼プログラムの登録は， 2 をキーインするとプログラムに必要なデータを次々に聞いてくるので，その指示に従ってキーインすればＯＫですが，コントロールの範囲が大ファイルになっているし，書き替えを行なうとファイルの項目の位置が変るので，あらかじめプログラムの簡単なフローチャートと項目を作って置いて登録すれば便利です．

プログラムの種類には，計算（Ｉ），メイン・サブ転換（Ｘ），縦計（Ｃ），横計（Ｌ），配列替え（Ａ），ソート（Ｓ）の６種類があります．

プログラムの式は，計算の場合は項目間の四則演算，％，巾（べき）計算式を，配列替えの場合は，項目の配列順および追加の記号を，ソートの場合は項目の記号に小さい順なら１，大きい順なら２をつけたものを指定します．

次の“W”は，この処理の結果に従ってファイルを書き替えるか否かを指定します．“Ｙ”なら書き替えます．次に新しくできる項目に名前を指定し，“CH”はその項目の桁数です．“・”は，小数点以下何桁を表示するかを数字で指定します．“P”は，この処理結果をプリンタに出力するか否かを指定します．“Ｙ”なら出力です．“C”は，連続計算をするか否かを指定します．“Ｙ”なら連続計算となり，次のNo.に継続するプログラム番号を指定します．

例　　自動計算プログラム・リスト

No.	S	プログラムメイ	プログラムノシキ	W	コウモクメイ	CH	・	P	C	No.
1	A	コンペケイサン	AB+	Y	Out	4		N	Y	2
2	A	～	ABC+	Y	In	4		N	Y	3
3	I	～	C+D	Y	Grs	5	0	N	Y	4
4	I	～	−0.8＊B+E	Y	Net	6	1	N	Y	5
5	S	～	F1	Y	Net sort			N	Y	6
6	I	～	C−D	Y	テンチ，オシドリ	9	0	Y	N	

⑧3をキーインするとEDITモードになります．不必要なプログラムを削除したり，プログラムの任意の部分を修正したりできます．

⑨4はリスト表示で修正した後の確認に使ってください．特に，連続演算や繰り返し演算の場合，プログラムを削除すると自動的にプログラムNo.がつまるので要注意．

⑩話が元に戻りますが，F7の演算命令にタテ計とヨコ計の機能があります．ヨコの合計は特に問題ありませんが，タテ計でファイルを書き替えると実はケイの上下に区切りの横線がデータとして追加されます．これは小計と合計を1度にやるために付けたもので，テープ・ベースでは小計機能を省略しているので盲腸的存在になっています．

これが実は配列替えや項目追加のときいたずらをします．ケイの下の横線データは通常は表示・印刷のとき出てこないようにしていますが，配列替などを行なうと表示の中に出てきます．したがって，演算や自動計算処理の場合はタテ計を一番最後にやるようにプログラムをしてください．

⑪CYRETをキーインすると，コントロール下のファイルをそのまま裏ファイルにコピーします．

これは，演算などを行なってデータが変わる前を保存した演算途中のカジュアル・バッファ・ファイルを必要とするときに使います．

◇参考文献

縄田，“汎用データ処理プログラムAUTO-2001”，I/O，1982.4

AUTO-2001 CMT-I プログラム・リスト

```
1 '**************************************
2 '   AUTO-2001 >>> CMT-1 <<<   V 1.0
3 'FM-8 VERSION PROGRAMMED BY COMPAC
4 '**************************************
5 CLEAR 5000,&H7FFF:DEFINTA:DEFINTC-M:DEFINTO-Z:WIDTH 80,25:CONSOLE 0,25,0,1:ON
ERROR GOTO8600
6 DIM D$(1,255),I$(1,26),S$(1,26),IC(1,26),IP(1,26),FD$(50),IN(1),IE(1),DE(1),F1
$(1),F2$(1),F3$(1),F4$(1),F5$(1),F6$(1),FT$(1),T$(1,26),T(1)
7 DIM X(26),Y(26),W(26):IC(1,1)=1
8 KEY 1,"ﾋｮ"+CHR$(13):KEY 2,"ｻﾌ"+CHR$(13):KEY 3,"ﾌﾟ"+CHR$(13):KEY 4,"ﾍﾝ"+CHR$(13
):KEY 5,"ｻｰ"+CHR$(13)
9 KEY 6,"ｸﾞ"+CHR$(13):KEY 7,"**"+CHR$(13):KEY 8,"**"+CHR$(13):KEY 9,"ﾃｰ"+CHR$(13
):KEY 10,"goto 99"+CHR$(13)
20 GOSUB7200
32 LOCATE 25,17:PRINT "ｱﾀﾗｼｸ ﾃﾞｰﾀ ｦ ﾂｸﾙﾄｷ --------1":LOCATE 25,20:PRINT "ﾏｴﾉ ﾃﾞｰﾀ
 ﾌｧｲﾙ ｦ ﾂｶｳﾄｷ ----2"
33 IF C3=1 THEN LOCATE 25,23:PRINT "( ﾌｧｲﾙ ｦ ｶｴﾙﾄｷﾊ f･2 ｦ ｷｰｲﾝ )":GOSUB 5980:BEE
P
34 A$=INKEY$:IF A$="" THEN 34
35 IF A$="ｻ" THEN IF F=0 THEN F=1:GOTO 32
36 IF A$="ｻ" THEN IF F=1 THEN F=0:GOTO 32
37 IF C3=1 THEN IF A$="g" THEN 99
38 A=VAL(A$):IF A>2 OR A=0 THEN GOSUB 5000:GOTO 34
39 IF A=2 THEN 82
40 LOCATE25,17:PRINTSPACE$(40):LOCATE25,20:PRINTSPACE$(40):LOCATE24,17:PRINT "年
月 日 ｦ ｲﾚﾃ ｸﾀﾞｻｲ｡":LOCATE28,19:INPUT"年 = ";A$:Z$=RIGHT$(" "+A$,2):C$=Z$+"/":LOCAT
E 62,0:PRINT C$:IF A$="goto 99" THENCLS:GOTO 32
41 LOCATE 28,20:INPUT"月 = ";A$:A$=RIGHT$(" "+A$,2):Z$=Z$+A$:C$=C$+A$+"/":LOCATE
62,0:PRINT C$:LOCATE 28,21:INPUT"日 = ";A$
42 A$=RIGHT$(" "+A$,2):Z$=Z$+A$:C$=C$+A$:LOCATE 62,0:C$=LEFT$(C$+SPACE$(30),8):P
RINT C$:LOCATE 20,23:INPUT "ﾔﾘﾅｵｼﾏｽｶ (Y/N) ";A$:LOCATE20,23:PRINTSPACE$(55):IFA$
<>"N" THEN CLS:GOTO32
44 LOCATE20,19:PRINTSPACE$(40):LOCATE0,20:PRINTSPACE$(40):LOCATE0,21:PRINTSPACE$
(40):LOCATE20,17:INPUT"ﾌｧｲﾙﾒｲ ( 6 ﾓｼﾞ ) = ";A$:IFA$="goto99"THEN32
45 FT$(F)=LEFT$(A$+SPACE$(6),6):LOCATE 8,0:PRINT FT$(F):LOCATE20,19:INPUT" ｺﾒﾝﾄ
( 15 ﾓｼﾞ ) = ";A$:A$=LEFT$(A$,15):IF LEFT$(A$,7)="goto 99"THEN 32
46 FT$(F)=LEFT$(FT$(F)+A$+SPACE$(15),21):LOCATE 8,0:PRINT LEFT$(FT$(F),6);"  ";R
IGHT$(FT$(F),15):FT$(F)=FT$(F)+Z$
47 LOCATE 20,23:INPUT"ﾔﾘﾅｵｼ ﾏｽｶ ? (Y/N) ";A$:IF A$="goto 99"THEN CLS:GOTO32
48 IF A$<>"N" THEN CLS:GOTO44
49 REM --------- ITEM SET ----------
50 DE(F)=0:WIDTH80:CONSOLE 13,12:LOCATE0,0:Y=12:IE(F)=0:F1$(F)="":F2$(F)="":F3$(
F)="":F4$(F)="":F5$(F)="":L$=""
51 GOSUB 5100:PRINT :PRINT TAB(22);">>> ｺｳﾓｸ ｾｯﾃｲ (ｵﾜﾘﾊ ] ｦ ｷｰｲﾝ) <<<":LOCATE 0,
10:PRINT STRING$(79,"-"):GOSUB 5980
52 LOCATE 22,9:PRINT "ｺｳﾓｸｶﾞ ﾓｼﾞﾉﾄｷﾊ ｢@ｺｳﾓｸﾒｲ｣ ﾄ ｽﾙｺﾄ"
54 LOCATE 22,11:PRINT "ｺｳﾓｸﾒｲ";TAB(45);"ｹﾀｽｳ(74ｹﾀﾏﾃﾞ)"
55 FOR I=1 TO 26
56 IF Y=23 THEN LOCATE 8,24:PRINT :Y=Y-1
57 LOCATE 8,Y+1:PRINT "ｺｳﾓｸ ( ";CHR$(64+I);" )   ";:INPUT A$:IF A$="]"THEN 71
60 IF LEFT$(A$,1)="@"THEN S$(F,I)="S":I$(F,I)=RIGHT$(A$,LEN(A$)-1):GOTO 64
62 S$(F,I)="N":I$(F,I)=A$:IF A$="goto 99"THEN 32
64 PRINT TAB(45);CHR$(30);:INPUT A$::IF A$="goto 99"THEN 32
65 IF ASC(A$)<48 OR ASC(A$)>57 THEN GOSUB 5000:GOTO 64
66 D=VAL(A$)+1
67 IC(F,I)=D:IE(F)=IE(F)+D:Y=Y+1:IF IE(F)>74 THEN GOSUB 5300:PRINT :PRINT TAB(9)
;"*** ｺｳﾓｸ ｹﾀｽｳ ﾉ ｵｰﾊﾞｰ ﾃﾞｽ !  ﾓｳｲﾁﾄﾞ ｾｯﾃｲ ｼﾃｸﾀﾞｻｲ ***":FOR J=0 TO 1000:NEXT:GOT
O 50
68 LOCATE 0,6:PRINT SPACE$(79):GOSUB 5400:LOCATE 0,5:PRINT " ┌───┬"+F1$(F):LOCATE
 0,6:PRINT " |No. |"+F2$(F):LOCATE 0,7:PRINT " ╞═══╪"+F3$(F);:IN(F)=I
69 LOCATE 60,11:PRINT SPACE$(18):LOCATE 60,11:X=73-IE(F):IF X<0 THEN X=X+1
70 PRINT " * ﾉｺﾘﾉｹﾀｽｳ =";X;" *";:NEXT I
71 BEEP 1:LOCATE 7,24:BEEP 0:PRINT " ﾓｼﾞ ｦ ﾃｲｾｲ ｽﾙ ---1   ｾｯﾃｲ ｦ ﾔﾘﾅｵｽ -----2  ｾ
ｯﾃｲ ｦ ｵﾜﾙ -------3"
72 A$=INKEY$:IF A$="" THEN 72
73 IF A$<"0" OR A$>"9" THEN GOTO 72
74 A=VAL(A$):IF A>3 OR A=0 THEN GOSUB 5000:GOTO 72
75 LOCATE 0,23:PRINT SPACE$(79)
76 IF A=2 THEN 50
78 IF A=3 THEN C3=0:CONSOLE8,17:GOTO 100
80 LOCATE 5,4:GOSUB 5200:GOSUB 5600:LOCATE 9,23:GOTO 71
81 REM ----- FILE LOAD -----
82 C3=1
```

CMT-I プログラム・リスト

```
83 CLS:GOSUB 5980:LOCATE 27,20:BEEP:INPUT"    ﾌｧｲﾙﾒｲ = ";FC$:FC$=LEFT$(FC$+"
",6):IF C1=1 THEN IF FC$="goto 9" THEN 1500
84 IF FC$="goto 9" THEN 32
85 IF FC$="ｦ     " THEN FC$=""
89 F1$(F)="":F2$(F)="":F3$(F)="":F4$(F)="":F5$(F)="":F6$(F)="":OPEN "I",#1,"CASO
:"+FC$:LOCATE30,22:PRINT FC$+" ｦ ﾖﾐｺﾐ ﾁｭｳ":FOR I=1 TO 26:INPUT #1,A$,B$,C$:IF A$
="I-END" THEN 91
90 I$(F,I)=A$:S$(F,I)=B$:IC(F,I)=VAL(C$):IE(F)=IE(F)+IC(F,I):GOSUB 5400:IN(F)=I:
NEXT I
91 FOR I=1 TO 255:INPUT #1,D$(F,I):IF D$(F,I)="D-END" THEN INPUT #1,FT$(F):DE(F)
=I-1:GOTO 93
92 NEXT I
93 FOR I=1 TO 26:INPUT #1,T$(F,I):IF T$(F,I)="C-END" THEN CLOSE #1:T(F)=I-1:GOTO
 96
94 NEXT I
95 S$(F,1)="S":I$(F,1)=LEFT$(LEFT$(FT$(F),6)+"  "+MID$(FT$(F),7,15)+SPACE$(70),6
1)+MID$(FT$(F),22,2)+"/"+MID$(FT$(F),24,2)+"/"+RIGHT$(FT$(F),2):D=70:GOTO 67
96 IF C1=1 THEN 98
98 CLS:CONSOLE8,17:GOTO155
99 GOSUB 7000:GOSUB 5500
100 IP(F,1)=1:F6$(F)="|   |":FOR I=1 TO IN(F)-1:IP(F,I+1)=IP(F,I)+IC(F,I):F6$(F)
=F6$(F)+SPACE$(IC(F,I)-1)+"|":NEXT I
102 F6$(F)=F6$(F)+SPACE$(IC(F,IN(F))-1)+"|"
109 IF C3=0 THEN C3=1:GOTO 200
111 GOSUB 5980
112 C1=1:LOCATE 70,0:BEEP 1:FOR N=0 TO 100:NEXT N:BEEP 0:A$=" ":INPUT A$:A$=LEFT
$(A$,2)
113 IF A$="go"THEN 98
114 IF A$="" THEN 150
120 I=INSTR("ﾋｮｻﾌﾌﾟﾍﾝﾃｰSZszｻｰｸﾞCLcl**",A$)
130 IF I=0 THEN 150
135 LOCATE0,4:PRINTSPACE$(79)
140 ON(I+1)/2 GOTO 400,600,800,1000,1500,180,180,2000,2500,3000,3000,145
145 CONSOLE0,25,0,0:KEY2,"LIST"+CHR$(13):KEY3,"RUN"+CHR$(13):KEY9,"LOAD":END
150 IF C2=1 THEN C2=0:GOTO99
155 LOCATE 0,2:PRINT "f1=ﾋｮｳｼﾞ       f2=ﾒｲﾝ<>ｻﾌﾞ      f3=ﾌﾟﾘﾝﾄ         f4=ﾍﾝｼｭｳ   f5=ｻｰﾁ"
156 LOCATE 0,3:PRINT "f6=ｸﾞﾗﾌ        f7= END         f8= END          f9=ﾃｰﾌﾟ    f10=ﾒｲﾚｲﾏﾁ
ﾍ";:C2=1
170 GOTO99
180 W=FRE(0):W=(W-560)/(IE(F)+5):W=INT(W):IF W<0 THEN W=0
182 GOSUB 7100:LOCATE 0,0:PRINT "ﾉｺﾘｻｲｽﾞ  | ｱﾄ";W;" ｷﾞｮｳ ｱﾘﾏｽ。          (ｽｷﾅ ｷｰ ｦ ｵｼ
ﾃｸﾀﾞｻｲ)"
184 A$=INKEY$:IF A$="" THEN 184
186 GOTO 99
199 REM ------- INPUT ------
200 CLS2:GOSUB5980:GOSUB5700:A$="1":IF DE(F)>0 THEN GOSUB 7100:LOCATE 0,0:PRINT
"ﾆｭｳﾘｮｸ| DL=ｻｸｼﾞｮ    ﾂｲｶ ﾊ <";DE(F)+1;" > ｷﾞｮｳ ｶﾗ。";:INPUT"     ﾜﾘｺﾐ ｷﾞｮｳﾊﾞﾝｺﾞｳ
";A$
205 IF A$="DL" OR A$="dl" THEN 250
210 LOCATE 0,0:PRINT " ﾆｭｳﾘｮｸ ":IF LEFT$(A$,2)="go" THEN 99
215 A=VAL(A$):IF A<1 OR A>DE(F)+1 THEN GOSUB 5000:GOTO 200
220 CLS1:LOCATE 0,4:PRINT ">>> ﾆｭｳﾘｮｸ <<<":GOSUB 5700:GOSUB 7100:LOCATE 9,0:PRIN
T " ｸｳﾗﾝﾊ ｢,｣ ｵﾜﾘﾊ ] ｦ ｷｰｲﾝ。     (｢:｣ﾋﾀﾞﾘ｢^｣ｳｴﾆ ｵﾅｼﾞ  ｢>｣ﾃｲｾｲ ｢<｣ﾋﾀﾞﾘｦｹｽ)"
221 Y=8
222 BEEP 1:FOR N=0 TO 100:NEXT :BEEP 0:IF DE(F)=255 OR FRE(0)<500 THEN GOTO 5800
224 A$="":B$="":C$="":IF F=1 THEN GOSUB 5980
225 FOR I=1 TO IN(F):LOCATE 0,Y:PRINT "|"+RIGHT$("    "+STR$(A),3)+"|";:TABX=IP(F
,I)+4:GOSUB 8000:IF S$(F,I)="S" THEN IF IC(F,I)=70 THEN GOSUB 5200:B$=MID$(CC$,6
,70):C$=X$:GOTO 234
226 IF S$(F,I)="S" THEN INPUT B$:GOTO 234
228 LINEINPUT C$
230 IF C$="goto 99" THEN 99
232 B$=RIGHT$(SPACE$(70)+C$,IC(F,I)-1)
234 IF LEFT$(B$,7)="goto 99" THEN 99
236 IF LEFT$(B$,1)=","OR C$=","THEN B$="    "
237 IF LEFT$(B$,1)="]" OR C$="]" THEN IF I=1 THEN 246
238 IF LEFT$(B$,1)="<" OR RIGHT$(B$,1)="<" OR RIGHT$(C$,1)="<" OR RIGHT$(X$,1)="
<" THEN PRINT CHR$(30);SPACE$(79);CHR$(30):GOTO 224
239 IF LEFT$(B$,1)=">" OR C$=">" THEN PRINT CHR$(30);SPACE$(79);CHR$(30):GOSUB 5
900:GOTO 224
240 IF LEFT$(B$,1)="^" OR C$="^" THEN B=0:FOR J=0 TO I-1:B=B+IC(F,J):NEXT J:B$=M
ID$(D$(F,A-1),B+1,IC(F,I))
241 IF LEFT$(B$,1)=":" OR C$=":" THEN B$=MID$(A$,IP(F,I-1),IC(F,I-1)-1):C$=""
243 A$=A$+LEFT$(B$+SPACE$(70),IC(F,I)-1)+"|":LOCATE 0,Y:PRINT "|"+RIGHT$("    "+S
TR$(A),3)+"|"+A$:NEXT I:Y=Y+1:IF Y=24 THEN PRINT :Y=Y-1
244 IF A=DE(F)+1 THEN D$(F,A)=A$:A=A+1:DE(F)=A-1:GOTO 222
245 FOR I=DE(F) TO A STEP -1:D$(F,I+1)=D$(F,I):NEXT I:D$(F,A)=A$:A=A+1:DE(F)=DE(
F)+1:GOTO 222
246 LOCATE 0,Y:GOTO 5150
250 LOCATE 0,0:PRINT " ｻｸｼﾞｮ  ":GOSUB 5250:IF A$="goto 99" THEN 99
252 A=DE(F):GOSUB 5750
254 A$=INKEY$:IF A$="" THEN 254
256 IF A$<>"y" AND A$<>"Y" THEN 99
258 IF L+M>DE(F) THEN DE(F)=L-1:GOTO 264
260 FOR I=L TO DE(F):PRINT I;CHR$(30):IF I=DE(F)+1-M THEN DE(F)=DE(F)-M:GOTO 264
262 D$(F,I)=D$(F,I+M):NEXT I
264 FOR I=DE(F)+1 TO A:PRINT I;CHR$(30):D$(F,I)="":NEXT I:GOTO 99
399 REM ------ display ------
400 LOCATE 0,0:PRINT "  ﾋｮｳｼﾞ ":IF F=1 THEN GOSUB 5980
424 GOSUB 5250 IF A$="goto 99" THEN 99
426 GOSUB 7100:PRINT "  ﾆﾝｲﾉ ｷｰｦ ｵｼﾃ ｽﾄｯﾌﾟｽﾀｰﾄ ｼﾏｽ。             (ｽﾄｯﾌﾟﾉﾄｷ ｢^｣=ﾃｲｾｲ,
f10=ﾒｲﾚｲﾏﾁ":Y=8
427 LOCATE 0,4:PRINT ">>> ﾋｮｳｼﾞ <<< ":CLS1
429 GOSUB 5700:FOR I=L TO L+M-1:IF I=DE(F)+1 THEN 5150
430 IF I=DE(F) AND RIGHT$(D$(F,I),1)="ｲ" THEN 5150
432 A$=INKEY$:IF A$="" THEN 438
434 A$=INPUT$(1)
435 IF A$="^" THEN GOSUB 5900:LOCATE 0,Y:PRINT SPACE$(79):GOTO 450
436 IF A$="g" THEN 5990
```

```
438 GOSUB 550:NEXT I:GOTO 5150
450 IF Y=24 THEN Y=Y-1:GOTO 434
452 GOTO 434
549 REM ----- disp sub -----
550 IF Y=24 THEN PRINT :Y=Y-1
555 LOCATE 0,Y:PRINT "|"+RIGHT$("    "+STR$(I),3)+"|"+D$(F,I):Y=Y+1:RETURN
600 IF F=0 THEN F=1:G=0:GOTO 99
610 F=0:G=1:GOTO 99
799 REM ----- printer -----
800 GOSUB 7100:LOCATE 0,0:PRINT " ｲﾝｻﾂ |  ***   ﾌﾟﾘﾝﾀｰ ﾉ ｽｲｯﾁ ﾊ ﾊｲｯﾃ ｲﾏｽｶ ?   *
**":GOSUB11000:PRINT#2,CHR$(27);"O";
805 GOSUB 7100:PRINT " ｶﾞﾒﾝﾌﾟﾘﾝﾄ --- 1 ｱﾃﾅ ｲﾝｻﾂ --- 2 ﾃﾞｰﾀ ｲﾝｻﾂ ---.3 "
810 A$=INKEY$:IF A$="" THEN 810
815 IF A$="g" THEN 99
820 A=VAL(A$):IF A>3 OR A=0 THEN GOSUB 5000:GOTO 810
825 IF A=3 THEN 878
830 IF A=1 THEN LOCATE 0,0:PRINT SPACE$(159):LOCATE 0,2:HARDC:GOTO 99
840 IF A=2 THEN GOSUB 7100:LOCATE 0,0:PRINT "ｱﾃﾅｲﾝｻﾂ | ｺｳﾓｸｶﾞ 「ﾅﾏｴ」,「ｼﾞｭｳｼｮ」,「ﾕｳ
ﾋﾞﾝﾊﾞﾝｺﾞｳ」 ﾉ ｼﾞｭﾝﾆ ﾅｯﾃｲﾏｽｶ (Y/N) ?";
850 A$=INKEY$:IF A$="" THEN 850
860 IF A$<>"Y" THEN 940
862 GOSUB 7100:PRINT " ｱﾃﾅ ｲﾝｻﾂ ﾖｳｼ ｦ ｾｯﾄ ｼﾃ ﾄﾞﾚｶ ｽｷﾅ ｷｰ ｦ ｵｼﾃ ｸﾀﾞｻｲ｡"
864 A$=INKEY$:IF A$="" THEN 864
866 IF A$="g" THEN 99
868 FOR I=1 TO DE(F):PRINT#2,MID$(D$(F,I),IP(F,3),IC(F,3)-1):PRINT#2,MID$(D$(F,I
),IP(F,2),IC(F,2)-1)
870 FOR J=1 TO IC(F,1):IF MID$(D$(F,I),IC(F,1)-J,1)>"O" THEN 874
872 NEXT J
874 PRINT#2:PRINT#2,TAB(5)MID$(D$(F,I),IP(F,1),IC(F,1)-J)+"  ｻﾏ":PRINT#2:PRINT#2
:NEXT I:GOTO 940
878 GOSUB 5250:IF A$="goto 99" THEN 940
881 GOSUB 7100:PRINT" ｷﾞｮｳｶﾝ ｦ ｱｹﾀ ｲﾝｻﾂ ﾆ ｼﾏｽｶ (Y/N) ?":Z$=INPUT$(1):IF Z$<>"Y"
THEN D=40:GOTO 887
882 D=80
887 K=0:GOSUB 7100:PRINT " ﾌｧｲﾙﾒｲ ﾋｯﾞｹ ｦｲﾝｻﾂ ｼﾏｽｶ (Y/N) ?":LOCATE0,2:PRINTSPACE$
(159);:LOCATE0,2:GOSUB 5100
888 A$=INKEY$:IF A$="" THEN 888
889 IF A$="Y" THEN PRINT#2,LEFT$(FT$(F),6);"  ";MID$(FT$(F),7,15);TAB(70);MID$(F
T$(F),22,2);"/";MID$(FT$(F),24,2);"/";RIGHT$(FT$(F),2):GOTO 891
890 GOTO 892
891 PRINT#2,STRING$(78,"￣"):PRINT#2:PRINT#2
892 A=0:A$=LEFT$(F1$(F),LEN(F1$(F))-1):B$=LEFT$(F3$(F),LEN(F3$(F))-1):C$=LEFT$(F
5$(F),LEN(F5$(F))-1)
893 PRINT#2,"┌───┬"+A$+"┐":PRINT#2,"|No.|"+F2$(F):PRINT#2,"╞═══╪"+B$+"╡":FOR I=L
 TO L+M-1:IF I=DE(F)+1 THEN 900
894 IF I=DE(F) AND RIGHT$(D$(F,I),1)="┤" THEN 900
895 IF Z$="Y" THEN PRINT#2,F6$(F)
896 PRINT#2,"|"+RIGHT$("          "+STR$(I),3)+"|"+D$(F,I):A=A+1:IF (A/D)=INT(A/D
) THEN GOSUB 905
898 NEXT I
900 PRINT#2,"└───┴"+C$+"┘":GOTO 940
905 PRINT#2,"└───┴"+C$+"┘":FOR I=0 TO 6:PRINT#2:NEXT C:PRINT#2,"┌───┬"+A$+"┐":PR
INT#2,"|No.|"+F2$(F):PRINT#2,"╞═══╪"+B$+"╡":RETURN
940 FORJ=0TO4:PRINT#2:NEXTJ:LOCATE0,2:PRINTSPACE$(159);:GOTO99
955 FOR C=0 TO 14:PRINT#2:NEXT C:PRINT#2,"     "+I$(F,I):PRINT#2,"     "+STRING$
(71,"￣"):PRINT#2:PRINT#2:RETURN
999 REM ----- editing -----
1000 GOSUB 7100:LOCATE 0,0:PRINT " ﾍﾝｼｭｳ  | ﾃｲｾｲ --- 1 ｱﾍﾟﾝﾄﾞ --- 2 ﾃﾞｰﾀ ﾂｲｶ､ｻｸｼ
ﾞｮ --- 3";
1010 A$=INKEY$:IF A$="" THEN 1010
1020 IF A$="g" THEN 99
1030 A=VAL(A$):IF A>3 OR A=0 THEN GOSUB 5000:GOTO 1010
1040 IF A=1 THEN 4000
1050 IF A=2 THEN 4100
1060 IF A=3 THEN 200
1500 GOSUB 7100:PRINT " ﾃｰﾌﾟ ﾆ ｾｰﾌﾞ ---1 ｱﾗﾀﾆ ﾃｰﾌﾟ ﾖﾘ ﾛｰﾄﾞ ---2 "
1510 A$=INKEY$:IF A$="" THEN 1510
1520 IF A$="g" THEN 99
1530 A=VAL(A$):IF A>2 OR A=0 THEN GOSUB 5000:GOTO 1510
1550 IF A=2 THEN GOSUB 5750:GOTO 1800
1560 CLS:GOSUB 7100:LOCATE 0,0:PRINT "  ｾｰﾌﾞ  | ｼﾀﾉ ﾌｧｲﾙﾒｲ ﾄ ﾋﾂﾞｹ ﾃﾞ ｾｰﾌﾞ ｼﾏｽｶ (
Y/N) ?":LOCATE0,5:GOSUB 5100:GOSUB 5980
1570 A$=INKEY$:IF A$="" THEN 1570
1580 IF A$="g" THEN 99
1590 IF A$="Y" THEN FC$=LEFT$(FT$(F),6):GOTO 1625
1600 LOCATE24,20:INPUT"ﾌｧｲﾙﾒｲ ( 6 ﾓｼﾞ) = ";A$:IF A$="goto 99" THEN 99
1610 FC$=LEFT$(A$+SPACE$(6),6):LOCATE25,21:INPUT"ｺﾒﾝﾄ ( 15 ﾓｼﾞ) = ";A$:FC$=LEFT$
(FC$+A$+SPACE$(15),21)
1620 LOCATE24,22:INPUT"ｻｸｾｲ 年 ( 2 ﾓｼﾞ) = ";A$:A$=RIGHT$("  "+A$,2):FC$=FC$+A$
1622 LOCATE24,23:INPUT"     月 ( 2 ﾓｼﾞ) = ";A$:A$=RIGHT$("  "+A$,2):FC$=FC$+A$
1624 LOCATE24,24:INPUT"     日 ( 2 ﾓｼﾞ) = ";A$:A$=RIGHT$("  "+A$,2):FC$=FC$+A$:FT
$(F)=FC$:CLS1
1625 LOCATE30,20:PRINT"ﾃｰﾌﾟ ﾍ ｾｰﾌﾞ ｼﾏｽ"
1630 OPEN"O",#1,"CASO:"+LEFT$(FC$,6):LOCATE 28,23:PRINT LEFT$(FC$,6)+" ｦ ｶｷｺﾐ ﾁｭ
ｳ"
1640 FOR I=1 TO IN(F):PRINT #1,I$(F,I),S$(F,I),STR$(IC(F,I)):NEXT I:PRINT #1,"I-
END","X","X"
1650 FOR I=1 TO DE(F):PRINT #1,D$(F,I):NEXT I:PRINT #1,"D-END":PRINT #1,FT$(F)
1654 IF T(F)=0 THEN 1656
1655 FOR J=1 TO T(F):PRINT #1,T$(F,J):NEXT J
1656 PRINT#1,"C-END":CLOSE#1
1660 CLS1:LOCATE0,6:PRINTSPACE$(79):PRINTSPACE$(79):GOTO99
1800 B$=INKEY$:IF B$="" THEN 1800
1804 IF B$<>"Y" THEN 99
1808 FOR I=0 TO DE(F):D$(F,I)="":NEXT I:IE(F)=0:F1$(F)="":F2$(F)="":F3$(F)="":F4
$(F)="":F5$(F)="":F6$(F)="":DE(F)=0
1809 GOTO 83
2000 CLS1:GOSUB 7100:LOCATE 0,0:PRINT "  ｻｰﾁ  | ﾓｼﾞ ﾏﾀﾊ ｽｳｼﾞﾚﾂ ｹﾝｻｸ ---1  ｼﾞｮｳｹﾝ
 ｹﾝｻｸ ---2":LOCATE 0,5:GOSUB 5050
```

CMT-I プログラム・リスト

```
2042 A$=INKEY$:IF A$="" THEN 2042
2044 IF A$="g" OR A$="G" THEN 99
2046 A=VAL(A$):IF A>2 OR A=0 THEN GOSUB 5000:GOTO 2042
2048 IF A=2 THEN 2100
2050 GOSUB 7100:INPUT" ﾓｼﾞﾚﾂ ﾏﾀﾊ ｽｳｼﾞﾚﾂ ";B$:IF B$="goto 99" THEN 99
2051 GOSUB 7100:PRINT" ﾌｧｲﾙ ｦ ｶｷｶｴ ﾏｽｶ (Y/N) ?"
2052 A$=INKEY$:IF A$="" THEN 2052
2053 IF A$="g" THEN 99
2054 IF A$="Y" THEN A=0
2060 GOSUB 7100:PRINT "  ";B$;" ｦ ｹﾝｻｸ ﾁｭｳ｡":LOCATE 0,4:PRINT ">>> ｹﾝｻｸ <<<  { "
;B$;" }":B=1:GOSUB 5700
2062 LOCATE0,8:FOR I=1 TO DE(F):FOR J=1 TO IE(F):IF MID$(D$(F,I),J,LEN(B$))=B$ T
HEN GOSUB 2092:GOTO 2066
2064 NEXT J:GOTO 2068
2066 IF A$="Y" THEN A=A+1:D$(F,A)=D$(F,I)
2068 IJK=CSRLIN:LOCATE0,IJK:PRINT I:LOCATE0,IJK:NEXT I
2069 IF A$="Y" THEN DE(F)=A
2071 GOTO 5150
2092 PRINT "|"+RIGHT$("   "+STR$(I),3)+"|"+D$(F,I):RETURN
2100 CLS2:L=1:Z$="ｼﾞｮｳｹﾝ --- ":B=1:GOSUB 7100:PRINT " C=15 ﾏﾀﾊ B<=200 ﾉﾖｳﾆ ｼﾞｮｳｹ
ﾝｦ､ ｵﾜﾙﾄｷﾊ J ｦｲﾚﾙ｡":LOCATE 0,5:GOSUB 5050
2102 LOCATE 0,3:PRINT SPACE$(79):PRINT SPACE$(77):LOCATE 0,3:PRINT Z$:INPUT"ｼﾞｮｳ
ｹﾝ ﾊ ";A$:IF A$="goto 99" THEN 99
2104 IF A$="J" THEN 2132
2106 K=1:C$=MID$(A$,2,1):B$=MID$(A$,2,2):A=ASC(LEFT$(A$,1))-64:IF A<1 OR A>26 TH
EN GOTO 2102
2108 IF B$="<>" THEN J=4:GOTO 2122
2110 IF B$="<=" OR B$="=<" THEN J=5:GOTO 2122
2112 IF B$=">=" OR B$="=>" THEN J=6:GOTO 2122
2114 IF C$="=" THEN J=1:GOTO 2124
2116 IF C$="<" THEN J=2:GOTO 2124
2118 IF C$=">" THEN J=3:GOTO 2124
2120 GOSUB 5000:GOTO 2120
2122 K=K+1
2124 K=K+1:Z$=Z$+A$+","
2126 X$=MID$(A$+SPACE$(20),K+1,IC(F,A)-1):IF S$(F,A)="N" THEN X$=RIGHT$(A$,LEN(A
$)-K):X$=RIGHT$(SPACE$(50)+X$,IC(F,A)-1):W(L)=1:GOTO 2130
2128 W(L)=0
2130 FD$(L)=X$:X(L)=A:Y(L)=J:L=L+1:GOTO 2102
2132 BEEP:GOSUB 7100:PRINT " ﾌｧｲﾙ ｦ ｶｷｶｴ ﾏｽｶ (Y/N) "
2134 Y$=INKEY$:IF Y$="" THEN 2134
2136 IF Y$="g" THEN 99
2137 IF Y$="Y" THEN C=0
2138 GOSUB 7100:PRINT "  ";Z$;" ｦ ｹﾝｻｸ ﾁｭｳ｡":LOCATE 0,3:PRINTSPACE$(79):PRINT "
 >>> ｹﾝｻｸ <<<  { ";Z$;" }":GOSUB 5700
2140 LOCATE0,8:FOR M=1 TO DE(F)+2:IJK=CSRLIN:LOCATE0,IJK:PRINT M:LOCATE0,IJK
2142 FOR N=1 TO L-1:I=X(N):X$=FD$(N):J=Y(N):W=W(N):IF M=DE(F)+1 THEN 2195
2144 Z$=MID$(D$(F,M),IP(F,I),IC(F,I)-1)
2146 ON J GOTO 2148,2158,2168,2176,2181,2186
2148 IF W=0 THEN 2154
2150 IF VAL(Z$)=VAL(X$) THEN 2191
2152 GOTO 2194
2154 IF Z$=X$ THEN 2191
2156 GOTO 2194
2158 IF W=0 THEN 2164
2160 IF VAL(Z$)<VAL(X$) THEN 2191
2162 GOTO 2194
2164 IF Z$<X$ THEN 2191
2166 GOTO 2194
2168 IF W=0 THEN 2174
2170 IF VAL(Z$) > VAL(X$) THEN 2191
2172 GOTO 2194
2174 IF Z$>X$ THEN 2191
2175 GOTO 2194
2176 IF W=0 THEN 2179
2177 IF VAL(Z$)<>VAL(X$) THEN 2191
2178 GOTO 2194
2179 IF Z$<>X$ THEN 2191
2180 GOTO 2194
2181 IF W=0 THEN 2184
2182 IF VAL(Z$)<=VAL(X$) THEN 2191
2183 GOTO 2194
2184 IF Z$<=X$ THEN 2191
2185 GOTO 2194
2186 IF W=0 THEN 2189
2187 IF VAL(Z$)>=VAL(X$) THEN 2191
2188 GOTO 2194
2189 IF Z$>=X$ THEN 2191
2190 GOTO 2194
2191 NEXTN
2192 PRINT "|"+RIGHT$("   "+STR$(M),3)+"|"+D$(F,M)
2193 IF Y$="Y" THEN C=C+1:D$(F,C)=D$(F,M)
2194 NEXT M
2195 IF Y$="Y" THEN DE(F)=C
2197 GOTO 5150
2500 CLS1:LOCATE 0,5:GOSUB 5050:GOSUB 7100:LOCATE 0,0:INPUT " ｸﾞﾗﾌ | (ｻｲｼｮﾉ ｺｳ
ﾓｸ ｶﾞ ﾐﾀﾞｼ ﾆ ﾅﾘﾏｽ)  ｸﾞﾗﾌ ﾆ ｽﾙ ｺｳﾓｸ ";C$:IF C$="goto 99" THEN 99
2520 K=ASC(C$)-64:IF K>IN(F) OR K<0 THEN GOSUB 5000:GOTO 2500
2530 GOSUB 5250:IF A$="goto 99" THEN 99
2540 GOSUB 7100:INPUT"(ﾋｮｳｼﾞ ﾉ ﾄﾁｭｳ f3=ｶﾞﾒﾝ ﾌﾟﾘﾝﾄ)  ﾌﾙ ｽｹｰﾙ ｦ ｲﾚﾃ ｸﾀﾞｻｲ｡ ";CC$:
IF CC$="goto 99" THEN 99
2550 D=LEN(CC$):N=VAL(CC$)
2560 W=K:A=INT((76-IC(F,1))/4)*4:CONSOLE4,21:CLS2:PRINT TAB(24);"***  ";I$(F,K);
" ﾉ ｸﾞﾗﾌ  ***":PRINT TAB(24);STRING$(IC(F,K)+17,"‾")
2590 PRINT TAB(IC(F,1)-1);"0";SPACE$(A/2-D);N/2:PRINT TAB(IC(F,1)+A-D-1);CHR$(30
);N
2600 PRINT TAB(IC(F,1)-1)"├";STRING$(A/4-1,"─");"+";STRING$(A/4-1,"─");"+";STRIN
G$(A/4-1,"─");"+";STRING$(A/4-1,"─");"┤";
2605 LOCATE0,4:FOR I=L TO L+M-1:IF I=DE(F)+1 THEN 2622
```

```
2607 PRINT TAB(IC(F,1)-1);"|";SPACE$(A/4-1);"|";SPACE$(A/4-1);"|";SPACE$(A/4-1);
"|";SPACE$(A/4-1);"|";CHR$(30);
2610 IF I=L THEN LOCATE 0,4
2612 A$=INKEY$:IF A$="" THEN 2619
2614 A$=INKEY$:IF A$="" THEN 2614
2615 IF A$="ｶ" THEN GOSUB 2810:COPY 1:FOR J=0 TO 4:LPRINT :NEXT J:GOTO 99
2619 PRINT LEFT$(D$(F,I),IC(F,1));:B=VAL(MID$(D$(F,I),IP(F,W),IC(F,W)-1)):IF B<0
 THEN ABSB=ABS(B):C=INT((ABSB/N)*A):PRINT STRING$(C,"=")+"(ﾏｲﾅｽ)"
2620 IF B>0 THEN C=INT((B/N)*A):PRINT STRING$(C,"=")
2621 NEXT I
2622 GOSUB 2810
2623 A$=INKEY$:IF A$="" THEN 2623
2624 IF A$="ｶ" THEN HARDC:GOTO99
2630 GOSUB 7100:LOCATE 28,0:PRINT" ｲﾝｻﾂ ｼﾏｽｶ (Y/N) ?":Y$=INPUT$(1):IF Y$<>"Y" TH
EN 2773
2700 GOSUB11000:PRINT #2,TAB(24);"***  ";I$(F,K);" ﾉ ｸﾞﾗﾌ  ***":PRINT#2,TAB(24);
STRING$(IC(F,K)+17,"￣"):PRINT#2:PRINT#2,CHR$(27);"A";CHR$(7)
2702 PRINT#2,TAB(IC(F,1)-1);"0";SPACE$(A/2-D);N/2;TAB(IC(F,1)+A-D-1);N
2710 PRINT#2,TAB(IC(F,1)-1);"├";STRING$(A/4-1,"─");"┼";STRING$(A/4-1,"─");"┼";ST
RING$(A/4-1,"─");"┼";STRING$(A/4-1,"─");"┤"
2720 FOR I=L TO L+M-1:IF I=DE(F)+1 THEN 2770
2730 PRINT#2,TAB(IC(F,1)-1);"|";SPACE$(A/4-1);"|";SPACE$(A/4-1);"|";SPACE$(A/4-1
);"|";SPACE$(A/4-1);"|";CHR$(13)
2750 PRINT#2,LEFT$(D$(F,I),IC(F,1));:B=VAL(MID$(D$(F,I),IP(F,W),IC(F,W)-1)):IF B
<0 THEN B=ABS(B):C=INT((B/N)*A):PRINT#2,STRING$(C,"=")+"(ﾏｲﾅｽ)";:GOTO 2765
2760 C=INT((B/N)*A):PRINT#2,STRING$(C,"=");
2765 IF IC(F,1)-1+A>=0 THEN PRINT#2,TAB(IC(F,1)-1+A);"|"
2767 NEXT I
2770 PRINT#2,TAB(IC(F,1)-1);"|";SPACE$(A/4-1);"|";SPACE$(A/4-1);"|";SPACE$(A/4-1
);"|";SPACE$(A/4-1);"|"
2771 PRINT#2,TAB(IC(F,1)-1);"└";STRING$(A/4-1,"─");"┴";STRING$(A/4-1,"─");"┴";ST
RING$(A/4-1,"─");"┴";STRING$(A/4-1,"─");"┘"
2772 FOR I=0 TO 4:PRINT#2:NEXT I:PRINT#2,CHR$(27);"O"
2773 CLS:CONSOLE8,17:GOTO99
2810 PRINT TAB(IC(F,1)-1);"└";STRING$(A/4-1,"─");"┴";STRING$(A/4-1,"─");"┴";STRI
NG$(A/4-1,"─");"┴";STRING$(A/4-1,"─");"┘":RETURN
3000 GOSUB 7100:LOCATE 0,0:PRINT "ｸﾘｱ  | ";:IF F=0 THEN PRINT "ﾒｲﾝ ";:GOTO 30
04
3002 PRINT "ｻﾌﾞ ";
3004 PRINT "ﾌｧｲﾙ ｦ ｸﾘｱ ---1  ｵｰﾙ ｸﾘｱ ----2  ｼﾞｮﾌﾞ ｦ ｵﾜﾙ ----3"
3010 A$=INKEY$:IF A$="" THEN 3010
3020 IF A$="g" THEN 99
3025 IF A$<"0" OR A$>"9" THEN GOTO 3010
3030 A=VAL(A$):IF A>3 OR A=0 THEN GOSUB 5000:GOTO 3010
3040 IF A=1 THEN LOCATE 0,0:PRINT SPACE$(79):GOTO 32
3050 IF A=2 THEN RUN
3060 GOTO145
3999 REM ----- utility -----
4000 GOSUB 7100:LOCATE0,2:PRINTSPACE$(54):PRINTSPACE$(79):PRINTSPACE$(79):LOCATE
 0,0:PRINT "ED      | ﾃｲｾｲ ﾊ  1 ｷﾞｮｳ ｺﾞﾄﾆ ｵｺﾅｲﾏｽ｡      f10=ﾒｲﾚｲﾏﾁﾍ"
4010 LOCATE 0,24:BEEP:IF IC(F,1)=70 THEN GOSUB 5200:GOSUB 5600:C$=CC$:GOSUB 5910
:GOTO 4030
4020 GOSUB 5905
4030 IF LEFT$(X$,7)="goto 99" THEN X$="":LOCATE0,4:PRINTSPACE$(9):GOTO 99
4032 CC$=C$:GOSUB 5600:LOCATE 0,0:GOTO 4000
4100 GOSUB 7100:LOCATE 0,0:PRINT " ｱﾍﾟﾝﾄﾞ | ﾃｰﾌﾟ  ﾌｧｲﾙ ｦ ﾂﾅｸﾞ ----1 ";:IF F=0 TH
EN PRINT "ｻﾌﾞ ";:G=1:GOTO 4102
4101 PRINT "ﾒｲﾝ ";:G=0
4102 PRINT "ﾌｧｲﾙ ｦ ﾂﾅｸﾞ ----2"
4104 A$=INKEY$:IF A$="" THEN 4104
4106 IF A$="g" THEN 99
4108 A=VAL(A$):IF A>2 OR A=0 THEN GOSUB 5000:GOTO 4104
4109 IF A=1 THEN 4200
4110 IFDE(F)=0THENIN(F)=IN(G):F1$(F)=F1$(G):F2$(F)=F2$(G):F3$(F)=F3$(G)ELSE4120
4115 FORI=1TOIN(F):IC(F,I)=IC(G,I):IP(F,I)=IP(G,I):S$(F,I)=S$(G,I):I$(F,I)=I$(G,
I):NEXTI:IE(F)=IE(G):F5$(F)=F5$(G)
4120 FOR I=DE(F)+1 TO DE(F)+DE(G):IF FRE(0)<500 THEN 5800
4121 PRINT I;CHR$(30):D$(F,I)=D$(G,I-DE(F)):NEXT I:DE(F)=DE(F)+DE(G):GOTO 98
4200 CLS:LOCATE 30,20:INPUT"ﾌｧｲﾙ ﾒｲ = ";FC$:IF FC$="goto 99" THEN 99
4205 FC$=LEFT$(FC$+"      ",6):IF FC$="ⓐ     "THEN FC$=""
4210 OPEN "I",#1,"CAS0:"+FC$:LOCATE30,22:PRINT FC$+" ｦ ﾖﾐｺﾐ ﾁｭｳ"
4220 FOR I=1 TO 26:INPUT #1,A$,B$,C$:IF A$="I-END" THEN I=26
4222 NEXT I
4225 FOR I=DE(F)+1 TO 255:INPUT #1,D$(F,I):IF D$(F,I)="D-END" THEN III=I:I=255
4230 NEXT I
4240 CLOSE #1:DE(F)=III-1:CLS1:GOTO 99
5000 FOR N=10 TO 1 STEP -1:BEEP 1:FOR M=0 TO N:NEXT:BEEP 0:NEXT N:RETURN
5050 PRINT " ┌───┬"+F1$(F):PRINT " |No. |"+F2$(F):PRINT " ╞═══╪"+F3$(F);:LOCATE 0,CS
RLIN+1:B$="    ":FOR I=1 TO IN(F):B$=B$+SPACE$(IC(F,I)-1)+CHR$(64+I):NEXT I
5060 PRINT B$:PRINT SPACE$(79):RETURN
5100 PRINT "Title : ";LEFT$(FT$(F),6);"  ";MID$(FT$(F),7,15);TAB(64);"Date : ";M
ID$(FT$(F),22,2)+"/"+MID$(FT$(F),24,2)+"/"+RIGHT$(FT$(F),2)
5110 PRINT STRING$(79,"￣"):RETURN
5150 PRINT " └───┴"+F5$(F)+CHR$(29)+"┘":GOTO 5990
5200 LINEINPUT;X$
5205 IF LEFT$(X$,7)="goto 99"THEN CC$=X$:RETURN
5210 CC$="":Y1=CSRLIN-1:FOR X=0 TO 79:B$=CHR$(SCREEN(X,Y1)):IF B$="" THEN B$=" "
5220 CC$=CC$+B$:NEXT X:RETURN
5250 GOSUB 7100:INPUT" ｢-｣=ｵｰﾙﾌｧｲﾙ｡      ｷﾞｮｳﾊﾞﾝｺﾞｳ (ﾊｼﾞﾒﾉ No)-(ｵﾜﾘﾉ No)  ? - ";A
$:IF A$="-" OR A$="." THEN L=1:M=255:RETURN
5252 IF A$="goto 99" THEN RETURN
5253 IF A$="" THEN 5250
5254 C=LEN(A$):FOR I=1 TO C:IF MID$(A$,I,1)<>"-" THEN NEXT
5255 IF C+1=I THEN A=VAL(A$):L=A:M=1:RETURN
5256 IF I=1 THEN A=VAL(A$):B=ABS(A):L=1:M=B:RETURN
5257 A=VAL(LEFT$(A$,I-1))
5258 L=A:I=C-I:B$=RIGHT$(A$,I):D=VAL(B$):M=D-A+1:IF B$="" THEN M=255:RETURN
5260 RETURN
5300 FOR N=100 TO 0 STEP -5:BEEP 1:FOR M=0 TO 20:NEXT M:BEEP 0:NEXT N:RETURN
```

CMT-1 プログラム・リスト

```
5400 F1$(F)=F1$(F)+STRING$(IC(F,I)-1,"-")+"┬":IF S$(F,I)="N" THEN F2$(F)=F2$(F)+
RIGHT$(SPACE$(70)+I$(F,I)+"|",IC(F,I)):GOTO 5420
5410 F2$(F)=F2$(F)+LEFT$(I$(F,I)+SPACE$(70),IC(F,I)-1)+"|"
5420 F3$(F)=F3$(F)+STRING$(IC(F,I)-1,"═")+"╪":F4$(F)=F4$(F)+STRING$(IC(F,I)-1,"-
")+"+":F5$(F)=F5$(F)+STRING$(IC(F,I)-1,"-")+"┴":RETURN
5500 IF IN(F)=0 THEN IN(F)=1:IC(F,1)=1
5510 A=1:FOR I=1 TO IN(F):I$(F,I)=MID$(F2$(F),A,IC(F,I)-1):A=A+IC(F,I):NEXT I:RE
TURN
5600 IF MID$(CC$+SPACE$(70),2,4)="No.|"THEN F2$(F)=MID$(CC$+SPACE$(70),6,IE(F)):
GOSUB 5500:RETURN
5610 RETURN
5700 LOCATE 0,5:PRINT "┌──┬"+F1$(F)+CHR$(29)+"┐":PRINT "|No.|"+F2$(F):PRINT "╞
══╪"+F3$(F)+CHR$(29)+"╡";:RETURN
5750 BEEP:LOCATE 9,0:PRINT "    *********** ﾎﾝﾄｳ ﾆ ｲｲﾃﾞｽｶ (Y/N) ? ***********
   ":RETURN
5800 GOSUB 7100:LOCATE 22,0:PRINT "******** ﾉｳﾘｮｸ ｲｯﾊﾟｲ ﾃﾞｽ｡ ********":GOSUB 5
300:FOR IIII=0 TO 1000:NEXT IIII:GOTO 99
5900 BEEP 1:LOCATE 0,Y:PRINT " ED ";:BEEP 0:IF IC(F,1)=70 THEN GOSUB 5200:C$=CC$
:GOTO 5910
5905 LINEINPUT C$
5906 IF LEFT$(C$,7)="goto 99" THEN LOCATE0,23:PRINTSPACE$(79):X$=C$:RETURN
5910 C=0:IF MID$(C$,5,1)="|"THEN C=VAL(MID$(C$,2,3))
5920 BEEP 1:D$(F,C)=MID$(C$+"    ",6,IE(F)):BEEP 0:RETURN
5980 IF F=0 THEN LOCATE 55,2:TABX=55:PRINT "    >>>>> ﾒｲﾝ ﾌｧｲﾙ <<<<<":RETURN
5982 LOCATE 55,2:PRINT "    ■■■■■ ｻﾌﾞﾌｧｲﾙ ■■■■■ ":RETURN
5990 IF F=1 THEN 5992
5991 GOTO 99
5992 GOSUB 7100:GOSUB 5980:LOCATE 0,1:PRINT STRING$(78,"‾")
5993 GOTO 99
7000 LOCATE 0,0:PRINT SPACE$(79):LOCATE 0,0:PRINT "♣ ﾒｲﾚｲﾏﾁ| ﾂｶｴﾙ ﾒｲﾚｲ = f1~f10,
, SZ=ﾉｺﾘ ｻｲｽﾞ, CL=ｸﾘｱ"
7010 LOCATE 0,1:PRINT STRING$(79,"‾"):IF C2=0 THEN LOCATE 0,2:PRINT SPACE$(79):L
OCATE 0,3:PRINT SPACE$(79);
7020 RETURN
7100 LOCATE 0,1:PRINT STRING$(79,"‾"):LOCATE 9,0:PRINTSPACE$(70):LOCATE9,0:RETUR
N
7200 LOCATE 21,8:PRINT "<<< AUTO-2001 CMT-1  Ver 1.0 >>>":LOCATE 24,11:PRINT "**
* Ube Tek Micom Club ***":RETURN
8000 HOR=CSRLIN:LOCATE 0,HOR:FOR ZET=1 TO TABX:PRINT CHR$(28);:NEXT ZET:RETURN
8600 CLOSE#1:IFERR=63 THENLOCATE40,20:PRINT"ﾌｧｲﾙ ｶﾞ ﾅｲ";CHR$(7):FORQQ=0TO1000:NE
XT:RESUME32
8610 IFERR=64 THENLOCATE25,24:PRINT"ｵﾅｼﾞ ﾅﾏｴﾉ ﾌｧｲﾙ ｶﾞ ｱﾙ";CHR$(7):RESUME1600
8620 RESUME99
11000 IFOLP=0THENOPEN"O",#2,"LPT0:":OLP=1:RETURNELSERETURN
```

AUTO-2001 CMT-3 プログラム・リスト

```
1 '************************************
2 '     AUTO-2001 >>> CMT-3 <<< V1.0
3 '   FM-8 VERSION PROGRAMMED BY COMPAC
4 '************************************
5 CLEAR 4500,&H7FFF:DEFINTB-M:DEFINTO-V:DEFINTX-Y:WIDTH 80,25:CONSOLE0,25,0,1
6 ON ERROR GOTO 8600
17 DIM D$(1,255),I$(1,26),S$(1,26),IC(1,26),IP(1,26),FD$(16),IN(1),IE(1),DE(1),F
1$(1),F2$(1),F3$(1),F4$(1),F5$(1),F6$(1),FT$(1),T$(1,26),T(1),X(26),Y(26),W(26),
U$(26),M$(26):IC(1,1)=1:OLP=0
18 KEY 1,"ﾋｮ"+CHR$(13):KEY 2,"ｻﾌ"+CHR$(13):KEY 3,"ｶﾞﾒﾝｺﾋﾟｰ"+CHR$(13):KEY 4,"ﾊｲ"+
CHR$(13):KEY 5,"ｿｰ"+CHR$(13)
19 KEY 6,"**"+CHR$(13):KEY 7,"ｹｲ"+CHR$(13):KEY 8,"ｼﾞ"+CHR$(13):KEY 9,"ﾃｰ"+CHR$(1
3):KEY 10,"goto 99"+CHR$(13)
21 F=0:G=1
22 GOSUB 7100:CLS:GOSUB 7200:GOSUB 5980:LOCATE 20,18:PRINT"( ﾛｰﾄﾞｽﾙ ﾌｧｲﾙ ｦ ｶｴﾙﾄｷ
ﾊ f.2 ｦ ｷｰｲﾝ )";CHR$(7)
23 LOCATE 25,20:INPUT"    ﾌｧｲﾙﾒｲ = ";FC$:FC$=LEFT$(FC$+"     ",6)
24 IF FC$="ｻﾌ    "THEN GOSUB 600:GOTO 22
25 IF FC$="@     "THEN FC$=""
83 F1$(F)="":F2$(F)="":F3$(F)="":F4$(F)="":F5$(F)="":F6$(F)="":OPEN "I",#1,"CAS0
:"+FC$:LOCATE 29,22:PRINT FC$;" ｦ ﾖﾐｺﾐ ﾁｭｳ":FOR I=1 TO 26:INPUT #1,A$,B$,C$:IF A
$="I-END" THEN 91
90 I$(F,I)=A$:S$(F,I)=B$:IC(F,I)=VAL(C$):IE(F)=IE(F)+IC(F,I):GOSUB 5400:IN(F)=I:
NEXT I
91 FOR I=1 TO 255:INPUT #1,D$(F,I):IF D$(F,I)="D-END" THEN INPUT#1,FT$(F):DE(F)=
I-1:GOTO 93
92 NEXT I
93 FOR I=1 TO 26:INPUT#1,T$(F,I):IF T$(F,I)="C-END" THEN CLOSE #1:T(F)=I-1:GOTO
97
94 NEXT I
97 CONSOLE8,17:GOTO155
99 FD$(11)="":FD$(14)="":GOSUB 7000:GOSUB 5500
111 GOSUB 5980
112 C1=1:LOCATE 70,0:BEEP:INPUT A$:A$=LEFT$(A$,2):IF A$="go"THEN 97
120 FOR I=1 TO 25 STEP 2:IF A$=MID$("ﾋｮｻﾌｶﾞﾊｲﾃｰｿｰｹｲｼﾞSZszCYcy**",I,2)THEN 140
130 NEXT I:GOTO 150
140 ON(I+1)/2 GOTO 400,170,175,1000,1500,2000,3000,8000,180,180,700,700,145
145 CONSOLE0,25,0,0:KEY2,"LIST"+CHR$(13):KEY3,"RUN"+CHR$(13):KEY9,"LOAD":IFOLP=1
THENPRINT#2,CHR$(27)+"O":ENDELSEEND
150 IF C2=1 THEN C2=0:GOTO99
155 LOCATE 0,2:PRINT"f.1=ﾋｮｳｼﾞ  f.2=ﾒｲﾝ<->ｻﾌﾞ  f.3=ｶﾞﾒﾝｺﾋﾟｰ   f.4=ﾊｲﾚﾂ  f.5=ｿｰﾄ"
:LOCATE 0,3:PRINT"f.6= END   f.7=ｹｲｻﾝ         f.8= ｼﾞﾄﾞｳ     f.9=ﾃｰﾌﾟ  f.10=ﾒｲﾚｲﾏﾁ
ﾍ";:C2=1:GOTO 99
170 GOSUB 600:GOTO 99
175 LOCATE 0,0:PRINT SPACE$(159):IFOLP=1THENPRINT#2,CHR$(27);"O"
176 HARDC:IFOLP=1THENPRINT#2,CHR$(27);"A";CHR$(7)
177 GOTO99
180 W=FRE(0):W=INT((W-3000)/(IE(F)+5)):IF W<0 THEN W=0
182 GOSUB 7100:LOCATE 0,0:PRINT"ﾉｺﾘｻｲｽﾞ | ｱﾄ";W;" ｷﾞｮｳ ｱﾘﾏｽ｡      (ｽｷﾅ ｷｰｦ ｵｼﾃｸﾀﾞ
ｻｲ)":A$=INPUT$(1):GOTO 99
```

```
399 REM ----- display -----
400 LOCATE 0,0:PRINT "  ヒョウシ" ":IF F=1 THEN GOSUB 5980
424 GOSUB 5250 IF A$="goto 99" THEN 99
425 GOSUB10000:IFFD$(11)="Y"THENGOSUB11100
426 GOSUB 7100:PRINT "  ニンイノ キーヲ オシテ ストップ゜スタート シマス。        (ストップ゜ノトキ 「^」=テイセイ,
f10=メイレイマチ":Y=8:FOR QQ=4 TO 7:LOCATE 0,QQ:PRINT SPC(79);:NEXT:CLS 1
427 LOCATE 0,4:PRINT ">>> ヒョウシ <<<"
428 IFFD$(11)="Y"THENPRINT#2," ┌──┬"+F1$(F):PRINT#2,"|No. |"+F2$(F):PRINT#2,"├══
╪"+F3$(F)
429 GOSUB 5700:FOR I=L TO L+M-1:IF I=DE(F)+1 THEN 439
430 IF I=DE(F) AND RIGHT$(D$(F,I),1)="┤" THEN 439
432 A$=INKEY$:IF A$="" THEN 438
434 A$=INPUT$(1)
435 IF A$="^" THEN GOSUB 5900:LOCATE 0,Y:PRINT SPACE$(79):GOTO 450
436 IF A$="g" THEN 5990
438 GOSUB 550:NEXT I
439 IFFD$(11)="Y"THENPRINT#2," └──┴"+F5$(F):FORI=1TO4:PRINT#2:NEXTI
440 GOTO5150
450 IF Y=24 THEN Y=Y-1:GOTO 434
452 GOTO 434
549 REM ----- disp sub -----
550 IF Y=24 THEN PRINT :Y=Y-1
555 LOCATE 0,Y:PRINT "|"+RIGHT$("  "+STR$(I),3)+"|"+D$(F,I)
556 IF FD$(11)="Y" THEN PRINT#2,F6$(F):PRINT#2,"|"+RIGHT$("   "+STR$(I),3)+"|"+D
$(F,I)
557 Y=Y+1:RETURN
600 IF F=0 THEN F=1:G=0:RETURN
610 F=0:G=1:RETURN
700 GOSUB 5750:A$=INPUT$(1):IF A$<>"Y" THEN 99
709 GOSUB 7100:PRINT"   *** ファイル ヲ コピー チュウ ***":G=0:IF F=0 THEN G=1
730 M=DE(G):FOR I=1 TO DE(F):IF FRE(0)<500 THEN 5800
740 BEEP 1:D$(G,I)=D$(F,I):BEEP 0:NEXT I:FOR I=1 TO 26:IF FRE(0)<500 THEN 5800
750 BEEP 1:IC(G,I)=IC(F,I):IP(G,I)=IP(F,I):I$(G,I)=I$(F,I):S$(G,I)=S$(F,I):T$(G,
I)=T$(F,I):BEEP 0:NEXT I
760 DE(G)=DE(F):F1$(G)=F1$(F):F2$(G)=F2$(F):F3$(G)=F3$(F):F4$(G)=F4$(F):F5$(G)=F
5$(F):F6$(G)=F6$(F)
770 IN(G)=IN(F):IE(G)=IE(F):FT$(G)=FT$(F):T(G)=T(F)
780 IF DE(G)<M THEN FOR I=DE(G)+1 TO M:D$(G,I)="":NEXT I
790 GOTO 99
1000 E=0:CLS:LOCATE0,1:PRINTSTRING$(79,"‾"):GOSUB 7100:GOSUB 5070:GOSUB 5980:LOC
ATE 0,0:INPUT" ハイレツガエ(ハイレツガエハ CbAdAB,ツイカハ AbcD+ ノヨウニスル。 ハイレツハ ";A$
1075 Z$="":K=0:FOR I=0 TO 26:W(I)=0:NEXT I:FOR I=0 TO 16:FD$(I)="":NEXT I
1080 IF A$="goto 99"THEN 99
1081 IF A$=""THEN 99
1090 A=LEN(A$):GOSUB 7100:PRINT" ファイル ヲ カキカエ マスカ (Y/N) ? ";
1091 Y$=INPUT$(1):IF Y$="g" GOTO 99
1101 GOSUB10000
1104 IF FD$(11)="Y" THEN GOSUB 5010
1106 E=0:J=1:N=0:FOR I=1 TO 26:CC$=MID$(A$,I,1):IF CC$="+"THEN 1110
1107 NEXT I
1108 N=1:GOTO 1120
1110 IF FD$(14)="A" THEN 1115
1113 GOSUB 7100:LOCATE 0,0:INPUT"コウモクツイカ | コウモクメイ ( モジ゛ノ トキ ハ 「@コウモクメイ」 トスル ) ";
C$:IF C$="goto 99" GOTO 99
1114 GOSUB 7100:INPUT"  ケタスウ ハ ";X$:IF X$="goto 99" GOTO 99
1115 D=VAL(X$):D=D+1
1116 L=IN(F)+1:IC(F,L)=D:IF LEFT$(C$,1)="@" THEN S$(F,L)="S":I$(F,L)=RIGHT$(C$,L
EN(C$)-1):GOTO 1118
1117 S$(F,L)="N":I$(F,L)=RIGHT$(SPACE$(40)+C$,IC(F,L)-1):IF FD$(14)="A" THEN 112
0
1118 GOSUB 7100:INPUT" (オナジ゛モノヲ イレルトキハ ソレヲ, コベ゛ツニ イレルトキハ @ ヲ キーイン)";FD$(8):IF FD
$(8)="goto 99" GOTO 99
1120 IF J=DE(F)+1 THEN IF Z$="*" THEN 1336
1121 IF J=DE(F)+1 THENIF(IN(F)>1)+(DE(F)=DE(G))THEN1225ELSE DE(F)=DE(G)
1122  CC$="":FOR I=1 TO A:X$=MID$(A$,I,1):IF X$=" " THEN I=A:GOTO 1195
1123 B$=MID$(A$,I,1):IF B$="+" THEN B=L:M=F:GOTO 1140
1124 IF ASC(B$)>96 THEN B=ASC(B$)-96:M=G:GOTO 1131
1130 B=ASC(B$)-64:M=F
1131 IF B<1 OR B>26 THEN GOSUB 5300:GOTO 99
1132 Z$=MID$(D$(M,J),IP(M,B),IC(M,B)):CC$=CC$+Z$
1140 IF J<>1 THEN 1195
1150 FD$(3)=FD$(3)+STRING$(IC(M,B)-1,"─")+"┬"
1160 IF S$(M,B)="N" THEN FD$(1)=FD$(1)+RIGHT$(SPACE$(79)+I$(M,B)+"|",IC(M,B)):GO
TO 1180
1170 FD$(1)=FD$(1)+LEFT$(I$(M,B)+SPACE$(79),IC(M,B)-1)+"|"
1180 FD$(2)=FD$(2)+STRING$(IC(M,B)-1,"═")+"╪":FD$(4)=FD$(4)+STRI"G$(IC(M,B)-1,"─
")+"┼"
1185 FD$(5)=FD$(5)+STRING$(IC(M,B)-1,"─")+"┴":FD$(6)=FD$(6)+STI    (IC(M,B)-1,"─
")+"┼":FD$(10)=FD$(10)+SPACE$(IC(M,B)-1)+"|"
1190 K=K+1:W(K)=IC(M,B):IF Y$="Y" THEN M$(K)=I$(M,B):U$(K)=S$(M,B)
1195 NEXT I:IF J<>1 AND E=1 THEN 1310
1197 IF J<>1 THEN 1207
1200 H=0:FOR I=1 TO K:H=H+W(I):NEXT I:IF (H>74)*(H<132) THEN Y$="N":GOSUB 7100:P
RINT" 74ケタヲ オーバ゛ー シテイマスノテ゛ ヒョウシ゛テ゛キマセン。";CHR$(7):E=1:IFFD$(11)="Y"THEN OLP1=1:GO
SUB11100:GOTO 1304ELSEGOTO1304
1202 IF H>131 THEN 5800
1203 GOSUB 7100:LOCATE 0,4:PRINT"  <<< ハイレツ ガエ >>>  [ ";A$;" ]":PRINTSPACE$(79)
:PRINTSPACE$(79):PRINTSPACE$(79);:CLS1
1204 LOCATE0,5:PRINT" ┌──┬";FD$(3);CHR$(29);"┐":PRINT"|No. |";FD$(1):PRINT"├══╪"
+FD$(2);CHR$(29);"╡";:LOCATE0,8
1205 IF FD$(11)="Y"THEN PRINT#2," ┌──┬"+FD$(3):PRINT#2,"|No. |"+FD$(1):PRINT#2,"├
══╪"+FD$(2)
1207 IF N=1 THEN 1220
1208 IF FD$(8)<>"@" THEN FD$(9)=FD$(8):GOTO 1212
1209 PRINT CHR$(7);"|";RIGHT$("        "+STR$(J),3);"|";CC$;:INPUT FD$(9):PRINT CHR
$(30);:IF FD$(9)="J" OR FD$(9)="goto 99" GOTO 99
1212 IF S$(M,B)="N" THEN CC$=CC$+RIGHT$(SPACE$(40)+FD$(9)+"|",D):GOTO 1220
1213 CC$=CC$+LEFT$(FD$(9)+SPACE$(40),D-1)+"|"
1220 PRINT "|";RIGHT$("       "+STR$(J),3);"|";CC$:IF Y$="Y" THEN D$(F,J)=CC$
```

CMT-3 プログラム・リスト

```
1221 IF FD$(11)="Y" THEN PRINT#2, "|   |"+FD$(10)
1222 IF FD$(11)="Y" THEN PRINT#2, "|"+RIGHT$("    "+STR$(J),3)+"|"+CC$:IF Y$="Y
" THEN D$(F,J)=CC$
1224 J=J+1:GOTO 1120
1225 PRINT"└───┴";FD$(6);CHR$(29);"┘"
1226 IF FD$(11)="Y" THEN PRINT#2,"└───┴"+FD$(6):FOR I=0 TO 4:PRINT#2:NEXT I
1227 IF Y$<>"Y"THEN 1250
1230 F1$(F)=FD$(3):F2$(F)=FD$(1):F3$(F)=FD$(2):F4$(F)=FD$(5):F5$(F)=FD$(6):IN(F)
=K:IE(F)=0:FOR I=1 TO K:I$(F,I)=M$(I):S$(F,I)=U$(I):IC(F,I)=W(I)
1240 IE(F)=IE(F)+IC(F,I):NEXT:IP(F,1)=1:FOR I=1 TO IN(F)-1:IP(F,I+1)=IP(F,I)+IC(
F,I):NEXT :IF FD$(14)="A" THEN 2900
1242 GOTO 99
1250 IF FD$(14)="A" GOTO 2900
1252 GOTO 99
1304 IF FD$(11)<>"Y" THEN FOR I=0 TO 5000:NEXT I:GOTO 99
1305 PRINT#2, "┌───┬"+FD$(3):PRINT#2,"|No.|"+FD$(1):PRINT#2,"╞═══╪"+FD$(2)
1310 IF N=1 THEN 1320
1312 IF FD$(8)<>"@" THEN FD$(9)=FD$(8):GOTO 1316
1314 CLS:GOSUB 7100:PRINT" カキコム ナイヨウ ヲ イレテ クダサイ。 ? ";CHR$(7):PRINT:INPUT FD$(9)
:IF FD$(9)="]" OR FD$(9)="goto 99" GOTO 99
1316 IF S$(M,B)="N" THEN CC$=CC$+RIGHT$(SPACE$(40)+FD$(9)+"|",D):GOTO 1320
1317 CC$=CC$+LEFT$(FD$(9)+SPACE$(40),D-1)+"|":GOTO 1320
1318 PRINT#2,"|"+RIGHT$("    "+STR$(J),3)+"|"+CC$
1320 PRINT#2,"|   |"+FD$(10)
1330 PRINT#2,"|"+RIGHT$("    "+STR$(J),3)+"|"+CC$
1335 J=J+1:Z$="*":GOTO 1120
1336 PRINT#2,"└───┴"+FD$(6);CHR$(29):FOR N=0 TO 4:PRINT#2:NEXT :GOTO 99
1499 '═══════TAPE═══════
1500 LOCATE0,0:PRINTSPACE$(80);:LOCATE0,0:PRINT "テープ ニ セーブ ---1 アラタニ テープ ヨリ ロ
ード ---2 "
1510 A$=INPUT$(1)
1520 IF A$="g" THEN 99
1530 A=VAL(A$):IF A>2 OR A=0 THEN GOSUB 5000:GOTO 1510
1550 IF A=2 THEN GOSUB 5750:GOTO 1800
1560 CLS:GOSUB 7100:LOCATE 0,0:PRINT "  セーブ  | シタノ ファイルメイ ト ヒヅケ デ セーブ シマスカ (
Y/N) ?":LOCATE 0,5:GOSUB 5100
1570 A$=INPUT$(1)
1580 IF A$="g" THEN 99
1590 IF A$="Y" THEN FC$=LEFT$(FT$(F),6):LOCATE26,21:PRINT"テープ ヘ セーブ シマス":GOTO
1630
1600 LOCATE 0,24:INPUT"     ファイルメイ ( 6モジ ) = ";A$:IF A$="goto 99" THEN 99
1610 FC$=LEFT$(A$+SPACE$(6),6):INPUT"     コメント ( 15 モジ ) = ";A$:FC$=LEFT$(FC$+A
$+SPACE$(15),21)
1620 INPUT"    サクセイ 年 ( 2 モジ ) = ";A$:A$=RIGHT$("  "+A$,2):FC$=FC$+A$
1622 INPUT"         月 ( 2 モジ ) = ";A$:A$=RIGHT$("  "+A$,2):FC$=FC$+A$
1624 INPUT"         日 ( 2 モジ ) = ";A$:A$=RIGHT$("  "+A$,2):FC$=FC$+A$:FT$(F)=FC$
1630 OPEN"O",#1,"CAS0:"+LEFT$(FC$,6):LOCATE7,24:PRINT LEFT$(FC$,6)+" ヲ カキコミ チュウ"
1640 FOR I=1 TO IN(F):PRINT #1,I$(F,I),S$(F,I),STR$(IC(F,I)):NEXT I:PRINT #1,"I-
END","X","X"
1650 FOR I=1 TO DE(F):PRINT #1,D$(F,I):NEXT I:PRINT #1,"D-END":PRINT #1,FT$(F)
1653 IF T(F)=0 THEN 1655
1654 FOR J=1 TO T(F):PRINT #1,T$(F,J):NEXT J
1655 PRINT #1,"C-END":CLOSE #1:CLS:GOTO 99
1800 B$=INKEY$:IF B$="" THEN 1800
1804 IF B$<>"Y" THEN 99
1808 GOTO 22
1999 '════SORT/SEARCH════
2000 CLS1:GOSUB 7100:LOCATE0,4:PRINTSPACE$(159);SPACE$(159);:LOCATE 0,5:GOSUB 50
50:LOCATE 0,0:PRINT "  ソート  |";:INPUT " ナラベカエル コウモク ";B$:IF B$="goto 99" GOTO
99
2202 I=ASC(B$)-64:IF (I>IN(F))+(I<1) THEN GOSUB 5000:GOTO 2000
2210 GOSUB 7100:PRINT " チイサイ シュン --- 1, オオキイ シュン --- 2 ";
2220 A$=INPUT$(1)
2230 IF A$="g" GOTO 99
2231 GOSUB 10000
2240 A=VAL(A$):IF (A<1)+(A>2) THEN GOSUB5000:GOTO 2000
2250 GOSUB 7100:LOCATE 11,0:PRINT B$;" ヲ ";:IF A=1 THEN PRINT"チイサイ"; ELSE PRINT"
オオキイ";
2270 PRINT " シュン ニ カキカエ チュウ !!"
2280 B=IP(F,I):C=IC(F,I):D=I:H=DE(F)+3*(LEFT$(D$(F,DE(F)),1)="-")
2282 IF (A=1)*(S$(F,D)="S") THEN 2320
2284 IF (A=2)*(S$(F,D)="S") THEN 2370
2285 IF (A=2)*(S$(F,D)="N") THEN 2350
2290 FOR J=1 TO H-1:LOCATE0,4:PRINT J
2300 A=VAL(MID$(D$(F,J),B,C-1)):W=J:FOR K=J+1 TO H
2305 IF A<VAL(MID$(D$(F,K),B,C-1)) GOTO 2315
2310 A=VAL(MID$(D$(F,K),B,C-1)):W=K
2315 NEXT:SWAP D$(F,J),D$(F,W)
2318 NEXT:GOTO 2420
2320 FOR J=1 TO H-1:LOCATE0,4:PRINT J:A$=MID$(D$(F,J),B,C-1):W=J:FOR K=J+1 TO H
2325 IF A$<MID$(D$(F,K),B,C-1) GOTO 2335
2330 A$=MID$(D$(F,K),B,C-1):W=K
2335 NEXT:SWAP D$(F,J),D$(F,W):NEXT:GOTO 2420
2350 FOR J=1 TO H-1:LOCATE0,4:PRINT J:A=VAL(MID$(D$(F,J),B,C-1)):W=J:FOR K=J+1 T
O H:IF A<=VAL(MID$(D$(F,K),B,C-1)) THEN A=VAL(MID$(D$(F,K),B,C-1)):W=K
2365 NEXT:SWAP D$(F,J),D$(F,W):NEXT:GOTO 2420
2370 FOR J=1 TO H-1:LOCATE0,4:PRINT J:A$=MID$(D$(F,J),B,C-1):W=J:FOR K=J+1 TO H
2375 IF A$=MID$(D$(F,K),B,C-1) GOTO 2390
2380 A$=MID$(D$(F,K),B,C-1):W=K
2390 NEXT:SWAP D$(F,J),D$(F,W):NEXT
2420 LOCATE0,4:PRINTSPACE$(79);:LOCATE 0,4:PRINT"   <<< ソート >>>   [ ";I$(F,D);"
]":CLS1:GOSUB 5700
2422 IF FD$(11)="Y" THEN PRINT#2,"┌───┬"+F1$(F):PRINT#2,"|No.|"+F2$(F):PRINT#2,"
╞═══╪"+F3$(F)
2430 LOCATE0,8: FOR I=1 TO DE(F):PRINT"|";RIGHT$("    "+STR$(I),3);"|";D$(F,I)
2431 IF FD$(11)="Y" THEN PRINT#2,F6$(F):PRINT#2,"|"+RIGHT$("    "+STR$(I),3)+"|
"+D$(F,I)
2432 NEXT I
2435 IF FD$(14)="S" THEN 2440
```

```
2436 IF FD$(11)="Y" THEN PRINT#2,"└───┴"+F5$(F):FOR I=0 TO 4:PRINT#2:NEXT
2437 GOTO 5150
2440 PRINT"└───┴";F5$(F);CHR$(29);"┘"
2441 IF FD$(11)="Y"THEN PRINT#2,"└───┴"+F5$(F):FOR I=0 TO 4:PRINT#2:NEXT
2442 GOTO 2900
2900 GOSUB 5500:IF FD$(12)="Y" THEN FD$(14)="":A$=FD$(13):GOTO 8155 ELSE GOTO 81
00
2999 '═════CALCULATE═════
3000 GOSUB 7100:LOCATE 0,0:PRINT" ｹｲｻﾝ | ｺｳﾓｸ ｶﾝ ｹｲｻﾝ --- 1, ﾀﾃ ﾉ ｺﾞｳｹｲ ---
2, ﾖｺ ﾉ ｺﾞｳｹｲ --- 3 ? ";
3010, A$=INPUT$(1)
3020 IF A$="g" GOTO 99
3030 A=VAL(A$):IF (A=0)+(A>3) THEN GOSUB 5000:GOTO 3010
3040 IF A=2 THEN 3400 ELSE IF A=3 THEN 3600
3052 IF F=0 THEN G=1 ELSE F=1:G=0
3060 CLS1:GOSUB 5070:GOSUB 7100:INPUT "A+1.2*b%B ﾉﾖｳﾆ ｼｷｦｲﾚﾙ(ﾂｶｴﾙｷｺﾞｳﾊ +-*/%^) "
;A$:IF A$="goto 99" GOTO 99
3070 A=LEN(A$):GOSUB 7100:PRINT" ﾌｧｲﾙ ｦ ｶｷｶｴﾏｽｶ (Y/N) "
3090 Y$=INPUT$(1):IF Y$="g" GOTO 99
3091 GOSUB 10000
3094 IF FD$(11)="Y" THEN GOSUB 5010
3100 GOSUB 7100:INPUT " ｺｳﾓｸ ﾒｲ ﾊ ";C$:IF C$="goto 99" GOTO 99
3110 GOSUB 7100:INPUT " ｹﾀｽｳ ﾊ ";X$:IF X$="goto 99" GOTO 99
3115 GOSUB 7100:INPUT " ｼｮｳｽｳ ｲｶ ﾅﾝｹﾀ ｦ ﾀﾞｼﾏｽｶ ";FD$(10):IF FD$(10)="goto 99" GO
TO 99
3116 W=VAL(FD$(10)):W=10^W:FD$(10)=STR$(W):W=VAL(FD$(10))
3120 D=VAL(X$):D=D+1:L=IN(F)+1:IC(F,L)=D:S$(F,L)="N":I$(F,L)=C$:J=1
3122 H=0:FORI=1TO L:H=H+IC(F,I):NEXTI
3124 IF(H>74)*(H<132)THENY$="N":SC=1:LOCATE9,0:PRINT" 74ｹﾀｦ ｵｰﾊﾞｰ ｼﾏｽ ﾉﾃﾞ ﾋｮｳｼﾞ
ﾃﾞｷﾏｾﾝ":IFFD$(11)="Y"THENOLP1=1:GOSUB11100
3126 IFH>131THEN5800
3130 IF J=DE(F)+1 GOTO 3960
3131 CC$="":Q=0:E=0:H=0:N=0:FD$(15)="":FOR I=1 TO A:B$=MID$(A$,I,1):IF B$=" " TH
EN I=30
3132 FOR Y=1 TO 6:IF B$=MID$("+-*/%^",Y,1) THEN Z$=B$
3134 NEXT Y:IF (ASC(B$)=46)+(ASC(B$)>47)*(ASC(B$)<58) THEN H=1:GOTO 3160
3136 IF (ASC(B$)>96)*(ASC(B$)<123) THEN B=ASC(B$)-96:M=G:GOTO 3170
3138 IF (ASC(B$)>64)*(ASC(B$)<91) THEN B=ASC(B$)-64:M=F:GOTO 3170
3150 IF Q=1 THEN N=VAL(CC$):CC$="":Q=0:H=0:GOTO 3200
3151 IF E=1 AND H=1 THEN N=-VAL(CC$):CC$="":H=0:E=0:GOTO 3200
3153 IF H=1 THEN Z=VAL(CC$):CC$="":B$=FD$(15):H=0:GOTO 3184
3154 IF (I=1)*(Z$="-") THEN E=1
3156 GOTO 3200
3160 FD$(15)=Z$:IF I=1 THEN Q=1
3162 IF I=A THEN Z=VAL(CC$+B$):B$=FD$(15):GOTO 3184
3166 CC$=CC$+B$:GOTO 3200
3170 Z=VAL(MID$(D$(M,J),IP(M,B),IC(M,B)-1)):IF I=1 THEN N=Z:GOTO 3200
3171 IF E=1 THEN N=-Z:W(49)=0:GOTO 3200 ELSE B$=Z$
3184 IF B$="+" THEN N=(N+Z)*W:N=INT(N+.5):N=N/W:GOTO 3200
3186 IF B$="-" THEN N=(N-Z)*W:N=INT(N+.5):N=N/W:GOTO 3200
3188 IF B$="*" THEN N=N*Z*W:N=INT(N+.5):N=N/W:GOTO 3200
3190 IF B$="/" AND Z=0 THEN N=0:GOTO 3200
3192 IF B$="/" THEN N=N/Z*W:N=INT(N+.5):N=N/W:GOTO 3200
3194 IF B$="%" AND Z=0 THEN N=0:GOTO 3200
3195 IF B$="%" THEN N=N/Z*W:N=INT(N*100+.5):N=N/W:GOTO 3200
3197 IF B$="^" AND N<0 THEN N=-N:N=N^Z*W:N=INT(N+.5):N=-N/W:GOTO 3200
3198 IF B$="^" THEN N=N^Z*W:N=INT(N+.5):N=N/W
3200 NEXT I:GOSUB 3904:GOTO 3130
3400 GOSUB 7100:LOCATE 0,0:PRINT "ﾀﾃﾉｺﾞｳｹｲ| (ｻｲｼｮﾉ ｺｳﾓｸｶﾞ ﾐﾀﾞｼﾄ ﾅﾘﾏｽ)   ﾌｧｲﾙ ｦｶｷ
ｶｴﾏｽｶ (Y/N) ? ";
3410 Y$=INPUT$(1):IF Y$="g" GOTO 99
3421 GOSUB 10000
3424 IF FD$(11)="Y" THEN GOSUB 5010
3425 GOSUB 7100:PRINT" ** ｹｲｻﾝﾁｭｳ **"
3430 LOCATE0,4:PRINTSPACE$(79):LOCATE0,4:PRINT"  <<< ﾀﾃ ﾉ ｺﾞｳｹｲ >>>":Y=8
3440 IF FD$(11)="Y" THEN PRINT#2,"┌───┬"+F1$(F):PRINT#2,"|No. |"+F2$(F):PRINT#2,"
╞═══╪"+F3$(F)
3441 GOSUB 5700:FOR I=1 TO DE(F):GOSUB 550:NEXT I
3444 C$="":FORI=1TO IN(F):C$=C$+STRING$(IC(F,I)-1,"─")+"┼":NEXTI
3445 C$=LEFT$(C$,LEN(C$)-1)+"┤"
3450 PRINT"|";RIGHT$("   "+STR$(DE(F)+1),3);"├";C$
3451 IF FD$(11)="Y" THEN PRINT#2,F6$(F):PRINT#2,"|"+RIGHT$("   "+STR$(DE(F)+1),3
)+"├"+C$:PRINT#2,F6$(F)
3460 FOR I=1 TO 26:W(I)=0:NEXT I:CC$=LEFT$("ｹｲ"+SPACE$(79),IC(F,1)-1)+"|":FOR J=
2 TO IN(F)
3470 FOR I=1 TO DE(F):W(J)=W(J)+VAL(MID$(D$(F,I),IP(F,J),IC(F,J)-1)):NEXT I
3472 IF S$(F,J)="S" THEN CC$=CC$+RIGHT$(SPACE$(79),IC(F,J)-1)+"|":GOTO 3476
3474 CC$=CC$+RIGHT$(SPACE$(79)+STR$(W(J)),IC(F,J)-1)+"|"
3476 NEXT J
3480 PRINT"|";RIGHT$("   "+STR$(DE(F)+2),3);"|";CC$
3481 IF FD$(11)="Y" THEN PRINT#2,"|"+RIGHT$("   "+STR$(DE(F)+2),3)+"|"+CC$:PRINT
#2,F6$(F)
3482 IF Y$<>"Y" THEN 3492
3490 D$(F,DE(F)+1)=C$:D$(F,DE(F)+2)=CC$:D$(F,DE(F)+3)=C$:DE(F)=DE(F)+3
3492 IF FD$(12)="Y" THEN I=VAL(FD$(13)):GOTO 3495
3493 IF FD$(11)="Y" THEN PRINT#2,"└───┴"+F5$(F):FOR N=0 TO 4:PRINT#2:NEXT
3494 GOTO 5150
3495 IF FD$(11)="Y" THEN PRINT#2,"└───┴"+F5$(F):FOR N=0 TO 4:PRINT#2:NEXT
3496 PRINT "└───┴"+F5$(F)+CHR$(29)+"┘"
3497 A$=FD$(13):GOTO 8155
3600 GOSUB 7100:LOCATE 0,0:PRINT"ﾖｺﾉｺﾞｳｹｲ| ﾌｧｲﾙ ｦ ｶｷｶｴ ﾏｽｶ (Y/N) ? ";
3610 Y$=INPUT$(1)
3615 IF Y$="g" GOTO 99
3616 GOSUB 10000
3619 IF FD$(11)="Y" THEN GOSUB 5010
3620 GOSUB 7100:INPUT" ｺｳﾓｸ ﾒｲ ﾊ ";C$:IF C$="goto 99" GOTO 99
3630 GOSUB 7100:INPUT" :ｹﾀｽｳ ﾊ ";X$:IF X$="goto 99" GOTO 99
3640 D=VAL(X$):D=D+1:L=IN(F)+1:IC(F,L)=D:S$(F,L)="N":I$(F,L)=C$:A$="ﾖｺﾉ ｺﾞｳｹｲ":J
=1
3642 H=0:FORI=1TO L:H=H+IC(F,I):NEXTI
```

CMT-3 プログラム・リスト

```
3644 IF(H>74)*(H<132)THENY$="N":SC=1:LOCATE9,0:PRINT" 74ｹﾀｦ ｵｰﾊﾞｰ ｼﾏｽ ﾉﾃﾞ ﾋｮｳｼﾞ
ﾃﾞｷﾏｾﾝ":IFFD$(11)="Y"THENOLP1=1:GOSUB11100
3646 IFH>131THEN5800
3650 IF J=DE(F)+1 THEN 3960
3660 N=0:FOR I=1 TO IN(F):N=N+VAL(MID$(D$(F,J),IP(F,I),IC(F,I)-1)):NEXT I:GOSUB
3904:GOTO 3650
3904 IF J<>1 THEN 3950
3905 FD$(1)=F1$(F)+STRING$(IC(F,L)-1,"─")+"┬"
3907 FD$(2)=F2$(F)+RIGHT$(SPACE$(79)+I$(F,L)+"│",IC(F,L))
3910 FD$(3)=F3$(F)+STRING$(IC(F,L)-1,"═")+"╪"
3911 FD$(4)=F4$(F)+STRING$(IC(F,L)-1,"─")+"┼"
3912 FD$(5)=F5$(F)+STRING$(IC(F,L)-1,"─")+"┴"
3913 IP(F,1)=1:F6$(F)="│   │":FOR I=1 TO IN(F)-1:IP(F,I+1)=IP(F,I)+IC(F,I):F6$(F
)=F6$(F)+SPACE$(IC(F,I)-1)+"│":NEXT I
3914 F6$(F)=F6$(F)+SPACE$(IC(F,IN(F))-1)+"│"
3915 F6$(F)=F6$(F)+SPACE$(IC(F,L)-1)+"│"
3940 GOSUB 7100:PRINT" ** ｹｲｻﾝﾁｭｳ **":FOR QQ=4 TO 7:LOCATE 0,QQ:PRINT SPC(78);:N
EXT:CLS 1
3945 LOCATE0,4:PRINT"  <<< ｹｲｻﾝ >>>     [ ";A$;" ]":IFSC=0THENPRINT"┌───┬";FD$(1)
;CHR$(29);"┐"
3946 IF FD$(11)="Y" THEN PRINT#2,"┌───┬"+FD$(1)
3947 IFSC=0THENPRINT"│No.│";FD$(2):PRINT"╞═══╪";FD$(3);CHR$(29);"╡";:LOCATE 0,CS
RLIN+1
3948 IF FD$(11)="Y" THEN PRINT#2,"│No.│"+FD$(2):PRINT#2,"╞═══╪"+FD$(3)
3950 IFLEFT$(D$(F,J),1)="-"THENX$=LEFT$(D$(F,J),LEN(D$(F,J))-1)+"+"+STRING$(IC(F
,L)-1,"─")+"┤"ELSEX$=D$(F,J)+RIGHT$(SPACE$(40)+STR$(N)+"│",IC(F,L))
3951 IF Y$="Y" THEN D$(F,J)=X$
3952 IFSC=0THENPRINT"│";RIGHT$("    "+STR$(J),3);"│";X$
3953 IF FD$(11)="Y"THEN PRINT#2,F6$(F)
3954 IF FD$(11)="Y"THEN PRINT#2,"│"+RIGHT$("    "+STR$(J),3)+"│"+X$
3955 J=J+1:RETURN
3960 IF FD$(11)="Y" THEN PRINT#2,"└───┴"+FD$(5):FOR N=0 TO 4:PRINT#2:NEXT
3961 IFSC=0THENPRINT"└───┴";FD$(5);CHR$(29);"┘"
3962 SC=0:IF Y$="Y" THEN F1$(F)=FD$(1):F2$(F)=FD$(2):F3$(F)=FD$(3):F4$(F)=FD$(4)
:F5$(F)=FD$(5):F6$(F)=FD$(6):GOTO 3965
3964 IF FD$(12)="Y" THEN A$=FD$(13):GOTO 8155 ELSE GOTO 99
3965 IN(F)=IN(F)+1:IE(F)=IE(F)+IC(F,L):IP(F,L)=IP(F,L-1)+IC(F,L-1):IF FD$(12)="Y
" THEN A$=FD$(13):GOTO 8155 ELSE 99
4999 '══════UTILITY══════
5000 PRINT CHR$(7):RETURN
5010 GOSUB 7100:PRINT"      ** ﾌﾟﾘﾝﾀｰ ｦ ｶｸﾆﾝ ｼﾃｸﾀﾞｻｲ **":GOSUB 11000:GOSUB11100:PRI
NT#2:RETURN
5050 PRINT"┌───┬";F1$(F):PRINT"│No.│";F2$(F):PRINT"╞═══╪";F3$(F);:B$="     ":FOR
I=1 TO IN(F):B$=B$+SPACE$(IC(F,I)-1)+CHR$(64+I):NEXT I
5060 LOCATE0,8:PRINT B$:RETURN
5070 LOCATE 68,1:X=73-IE(F):IF X<0 THEN X=X+1
5071 PRINT"(ﾉｺﾘ";X;")"
5072 LOCATE 0,5:GOSUB 5050:LOCATE 0,8:IF F=0 THEN PRINT"-ﾒｲﾝ-" ELSE PRINT"-ｻﾌﾞ-"
5074 LOCATE 0,9:PRINT"┌───┬";F1$(G):PRINT"│No.│";F2$(G):PRINT"╞═══╪";F3$(G):B$="
    ":FOR I=1 TO IN(G):B$=B$+SPACE$(IC(G,I)-1)+CHR$(96+I):NEXT I
5076 PRINT B$:LOCATE 0,12:IF G=0 THEN PRINT "-ﾒｲﾝ-" ELSE PRINT "-ｻﾌﾞ-"
5078 RETURN
5100 PRINT "Title : ";LEFT$(FT$(F),6);"   ";MID$(FT$(F),7,15);TAB(64);"Data : ";M
ID$(FT$(F),22,2);"/";MID$(FT$(F),24,2);"/";RIGHT$(FT$(F),2)
5110 PRINT STRING$(79,"￣"):RETURN
5150 PRINT "└───┴";F5$(F);CHR$(29);"┘":GOTO 99
5250 GOSUB 7100:INPUT " 「-」=ｵｰﾙ ﾌｧｲﾙ。     ｷﾞｮｳﾊﾞﾝｺﾞｳ (ﾊｼﾞﾒﾉ No)-(ｵﾜﾘﾉ No)  ? - "
;A$:IF A$="-" OR A$="." THEN L=1:M=255:RETURN
5252 IF A$="GOTO99" THEN RETURN
5254 C=LEN(A$):FOR I=1 TO C:IF MID$(A$,I,1)<>"-"THEN NEXT
5255 IF C+1=I THEN A=VAL(A$):L=A:M=1:RETURN
5256 IF I=1 THEN A=VAL(A$):B=ABS(A):L=1:M=B:RETURN
5257 A=VAL(LEFT$(A$,I-1))
5258 L=A:I=C-I:B$=RIGHT$(A$,I):D=VAL(B$):M=D-A+1:IF B$="" THEN M=255
5260 RETURN
5300 PRINT CHR$(7):RETURN
5400 F1$(F)=F1$(F)+STRING$(IC(F,I)-1,"─")+"┬":IF S$(F,I)="N" THEN F2$(F)=F2$(F)+
RIGHT$(SPACE$(70)+I$(F,I)+"│",IC(F,I)):GOTO 5420
5410 F2$(F)=F2$(F)+LEFT$(I$(F,I)+SPACE$(70),IC(F,I)-1)+"│"
5420 F3$(F)=F3$(F)+STRING$(IC(F,I)-1,"═")+"╪":F4$(F)=F4$(F)+STRING$(IC(F,I)-1,"─
")+"┼":F5$(F)=F5$(F)+STRING$(IC(F,I)-1,"─")+"┴":RETURN
5500 IF IN(F)=0 THEN IN(F)=1:IC(F,1)=1
5510 A=1:FOR I=1 TO IN(F):I$(F,I)=MID$(F2$(F),A,IC(F,I)-1):A=A+IC(F,I):NEXT
5512 IP(F,1)=1:F6$(F)="│   │":FOR I=1 TO IN(F)-1:IP(F,I+1)=IP(F,I)+IC(F,I):F6$(F
)=F6$(F)+SPACE$(IC(F,I)-1)+"│":NEXT I
5514 F6$(F)=F6$(F)+SPACE$(IC(F,IN(F))-1)+"│":RETURN
5700 LOCATE0,5:PRINTSPACE$(79);SPACE$(159);:LOCATE 0,5:PRINT"┌───┬";F1$(F);CHR$(
29);"┐":PRINT"│No.│";F2$(F):PRINT"╞═══╪"+F3$(F);CHR$(29);"╡";:RETURN
5750 PRINT CHR$(7):LOCATE 9,0:PRINT"    *********** ﾎﾝﾄｳ ﾆ ｲｲﾃﾞｽｶ (Y/N) ? ******
*****       ";:RETURN
5800 GOSUB 7100:LOCATE 22,0:PRINT"******** ﾉｳﾘｮｸ ｲｯﾊﾟｲ ﾃﾞｽ。 ********":GOSUB 53
00:FOR N=0 TO 5000:NEXT I:GOTO 99
5900 BEEP 1:LOCATE 0,Y:PRINT " ED ";:BEEP 0
5905 INPUT C$
5906 IF LEFT$(C$,7)="goto 99" THEN X$=C$:RETURN
5910 C=0:IF MID$(C$,5,1)="│"THEN C=VAL(MID$(C$,2,3))
5920 BEEP 1:D$(F,C)=MID$(C$+"    ",6,IE(F)):BEEP 0:RETURN
5980 IF F=0 THEN LOCATE 58,2:TABX=55:PRINT ">>>>> ﾒｲﾝ ﾌｧｲﾙ <<<<<":RETURN
5982 LOCATE 58,2:PRINT "■■■■  ｻﾌﾞﾌｧｲﾙ  ■■■■  ":RETURN
5990 IF F=1 THEN 5992
5991 GOTO 99
5992 GOSUB 7100:GOSUB 5980:LOCATE 0,1:PRINT STRING$(78,"￣")
5993 GOTO 99
7000 LOCATE 0,0:PRINT SPACE$(79):LOCATE 0,0:PRINT"♠ ﾒｲﾚｲﾏﾁ● ﾂｶｴﾙ ﾒｲﾚｲ = f.1-f.10
, SZ=ﾉｺﾘ ｻｲｽﾞ, CY=ﾌｧｲﾙ ｺﾋﾟｰ"
7010 LOCATE 0,1:PRINT STRING$(79,"￣"):IF C2=0 THEN LOCATE 0,2:PRINT SPACE$(79):L
OCATE 0,3:PRINT SPACE$(77);
7012 RETURN
7100 LOCATE 9,0:PRINT SPACE$(70):LOCATE 9,0:RETURN
```

```
7200 LOCATE 19,8:PRINT "<<<  AUTO-2001  CMT-3   Ver 1.0  >>>":LOCATE 24,11:PRINT
 "*** Ube Tek Micom Club ***":RETURN
8000 CLS
8001 LOCATEO,0:PRINTSPACE$(79);:LOCATE 0,0:PRINT" ｼﾞﾄﾞｳ  |  ** ﾌﾟﾛｸﾞﾗﾑ ﾘｽﾄ **"
8010 LOCATEO,6:PRINT"No. S ﾌﾟﾛｸﾞﾗﾑ ﾒｲ  ﾌﾟﾛｸﾞﾗﾑ ﾉ ｼｷ";SPC(19);"W ｺｳﾓｸﾒｲ      CH .
P C No.":PRINT STRING$(75,"-");:Y=9:
8015 CLS1:IF T(F)=0 THEN 8060
8020 FOR II=1 TO T(F)
8025 A$=INKEY$:IF A$="" GOTO 8030
8026 A$=INPUT$(1)
8028 IF A$="g" GOTO 8100
8030 I=II:A$=MID$(STR$(I)+"  ",2,2)+"| ":LOCATE 0,Y:PRINT A$+T$(F,I)
8050 Y=Y+1:NEXT
8060 LOCATEO,0:PRINT STRING$(75,"-");CHR$(7)
8100 GOSUB 7100:PRINT"ﾌﾟﾛｸﾞﾗﾑ ﾉ ｼﾞｯｺｳ ---1,ﾌﾟﾛ ﾉ ﾄｳﾛｸ ---2,ﾌﾟﾛ ﾉ ﾃｲｾｲ ---3,ﾌﾟﾛｸﾞ
ﾗﾑﾘｽﾄ ---4";
8110 A$=INPUT$(1)
8120 IF A$="g" THENCLS:GOTO 99
8130 IF VAL(A$)<1 THEN 8110 ELSE ON VAL(A$) GOTO 8154,8200,8400,8000
8135 GOSUB 5000:GOTO 8110
8154 P=F:GOSUB 7100:INPUT"  ﾌﾟﾛｸﾞﾗﾑ No. = ";A$:IF A$="goto 99" THEN99ELSECLS1:LO
CATEO,6:PRINTSPACE$(79):PRINTSPACE$(79);
8155 I=VAL(A$):A$=MID$(T$(P,I),15,30):Y$=MID$(T$(P,I),46,1):C$=MID$(T$(P,I),48,1
0):X$=MID$(T$(P,I),59,2)
8156 FD$(10)=MID$(T$(P,I),62,1):FD$(11)=MID$(T$(P,I),64,1):FD$(12)=MID$(T$(P,I),
66,1):FD$(13)=RIGHT$(T$(P,I),2):FD$(14)=LEFT$(T$(P,I),1):IF FD$(11)="Y" THEN GOS
UB 5010
8157 IF FD$(14)="I" THEN A=30:GOTO 3116 ELSE IF FD$(14)="C"THEN 3425
8159 IF FD$(14)="L" THEN 3640 ELSE IF FD$(14)="A" THEN Z$="":A=30:K=0:FOR I=1 TO
 10:FD$(I)="":NEXT I:FD$(8)="@":FOR I=1 TO 26:W(I)=0:NEXT I:GOTO 1106
8161 IF FD$(14)="S" THEN B$=LEFT$(A$,1):A$=MID$(A$,2,1):I=ASC(B$)-64:GOTO 2240
8162 IF FD$(14)="X"THEN X=F:F=G:G=X:A$=FD$(13):GOTO8155
8163 GOSUB 5300:GOTO 99
8200 LOCATE 0,Y:PRINT SPACE$(79):LOCATE 0,Y:FOR I=Y-8 TO 26:A$=MID$(STR$(I)+"  "
,2,2)+"|":PRINT A$
8210 GOSUB 7100:LOCATE 0,0:BEEP:INPUT " ﾄｳﾛｸ  | ]=ｵﾜﾘ｡ ﾌﾟﾛｸﾞﾗﾑ ﾉ ｼｭﾙｲ(ｹｲｻﾝ=I,ﾀﾃｹ
ｲ=C,ﾖｺｹｲ=L,ﾊｲﾚﾂ=A,ｿｰﾄ=S,ﾌｧｲﾙｷﾘｶｴ=X)";A$:IF A$="]" THEN T(F)=I-1:GOTO8100
8218 T$(F,I)=LEFT$(A$+" ",1)+" ":LOCATE 4,Y:PRINT A$
8220 GOSUB 7100:BEEP:INPUT "ﾌﾟﾛｸﾞﾗﾑ ﾉ ﾅﾏｴ (10ﾓｼﾞ) ";A$
8228 T$(F,I)=T$(F,I)+LEFT$(A$+SPACE$(10),10)+" ":LOCATE 6,Y:PRINT A$
8230 GOSUB 7100:BEEP:INPUT"ﾌﾟﾛｸﾞﾗﾑ ﾉ ｼｷ (30ﾓｼﾞ) ";A$
8238 T$(F,I)=T$(F,I)+LEFT$(A$+SPACE$(30),30)+" ":LOCATE 18,Y:PRINT A$
8240 GOSUB 7100:BEEP:INPUT"ﾌｧｲﾙ ｦ ｶｷｶｴ ﾏｽｶ (Y/N) ";A$
8248 T$(F,I)=T$(F,I)+LEFT$(A$+" ",1)+" ":LOCATE 49,Y:PRINT A$
8250 GOSUB 7100:BEEP:INPUT" ｺｳﾓｸ ﾒｲ (10ﾓｼﾞ) ";A$
8258 T$(F,I)=T$(F,I)+RIGHT$(SPACE$(10)+A$,10)+" ":LOCATE 51,Y:PRINT A$
8260 GOSUB 7100:BEEP:INPUT" ｹﾀｽｳ ";A$
8261 IF A$<"0" AND A$>"9" THEN 8260
8268 T$(F,I)=T$(F,I)+LEFT$(A$+"  ",2)+" ":LOCATE 62,Y:PRINT A$
8270 GOSUB 7100:BEEP:INPUT " ｼｮｳｽｳ ｲｶ ﾅﾝｹﾀ ﾀﾞｼﾏｽｶ ";A$
8278 T$(F,I)=T$(F,I)+LEFT$(A$+" ",1)+" ":LOCATE 65,Y:PRINT A$
8280 GOSUB 7100:BEEP:INPUT" ｲﾝｻﾂ ｼﾏｽｶ (Y/N) ";A$
8288 T$(F,I)=T$(F,I)+LEFT$(A$+" ",1)+" ":LOCATE 67,Y:PRINT A$
8290 GOSUB 7100:BEEP:INPUT" ﾚﾝｿﾞｸ ｹｲｻﾝ ｼﾏｽｶ (Y/N) ";A$
8298 T$(F,I)=T$(F,I)+LEFT$(A$+" ",1)+" ":LOCATE 69,Y:PRINT A$
8300 GOSUB 7100:BEEP:INPUT" ﾂﾂﾞｹﾙ ﾌﾟﾛｸﾞﾗﾑ No. ";A$
8301 IF A$<"0" AND A$>"9" THEN 8300
8308 T$(F,I)=T$(F,I)+LEFT$(A$+"  ",2):LOCATE 71,Y:PRINT A$:Y=Y+1:NEXT I
8400 GOSUB 7100:INPUT" ( ﾃｲｾｲ=ED )   ｻｸｼﾞｮ ｽﾙ ﾌﾟﾛｸﾞﾗﾑ No.= ";A$
8405 IF A$="ed" OR A$="ED" GOTO 8500
8410 IF A$="goto 99" THEN LOCATEO,4:PRINTSPACE$(7):GOTO 99
8420 A=VAL(A$):IF A=<0 THEN 8400
8425 GOSUB 5750
8426 A$=INPUT$(1):IFA$<>"Y"THEN 8100
8430 FOR I=A TO 25:T$(F,I)=T$(F,I+1):NEXT I:T(F)=T(F)-1:GOTO 8001
8500 GOSUB 7100:LOCATE 0,0:PRINT"ED        | ﾃｲｾｲ ﾊ 1 ｷﾞｮｳ ｺﾞﾄﾆ ｵｺﾅｲﾏｽ｡";CHR$(7):L
OCATE 0,8:LINE INPUT C$
8520 IF LEFT$(C$,7)="goto 99"THEN 8001
8530 IF MID$(C$,3,1)="|"THEN C=VAL(LEFT$(C$,2)):GOTO 8550 ELSE 8001
8550 T$(F,C)=MID$(C$,5,69):GOTO 8500
8600 IF ERR=63 THEN CLOSE:LOCATE 40,20:PRINT "ﾌｧｲﾙ ｶﾞ ﾅｲ";CHR$(7):FOR QQ=0 TO 10
00:NEXT:RESUME 22
8610 IF ERR=64 THEN CLOSE:LOCATE 25,24:PRINT "ｵﾅｼﾞ ﾅﾏｴ ﾉ ﾌｧｲﾙ ｶﾞ ｱﾙ";CHR$(7):RES
UME 1600
8620 RESUME 97
10000 GOSUB7100:PRINT" ｲﾝｻﾂ ｼﾏｽｶ (Y/N) ?";
10010 FD$(11)=INPUT$(1)
10020 IF FD$(11)="g" THEN RETURN99
10031 IF FD$(11)="Y"THEN 11000
10040 RETURN
11000 IF OLP=0 THEN OPEN"O",#2,"LPTO:":PRINT#2,CHR$(27);"A";CHR$(7);:OLP=1
11010 RETURN
11100 IFOLP1=0THENPRINT#2,CHR$(18);ELSEPRINT#2,CHR$(15)
11110 OLP1=0:RETURN
```

FM-8

ディスケット・ドクター



■COMPAC

大量のデータを処理するにはどうしても，フロッピーディスクが必要です．ディスクを使用していて，操作を間違えたり，プログラムの暴走によって，ディスクを壊してしまうと大変です．

こういう場合でも，元のデータはかなり残っており，ある程度は復元できます．

このプログラムは，FM-8のディスクを調べ，場合によっては修正するものです．

F-BASICディスクの構成

片面＝0～39トラック
トラック＝1～32セクタ
セクタ＝256バイト
クラスタ＝8セクタ

F-BASICで使用するフロッピーディスクのトラック1には，ディスク管理の情報が書かれています．

トラック1，セクタ1……FAT(File Allocation Table)
トラック1，セクタ4～32……ディレクトリ

FATはクラスタの使用状況を示すテーブルで，5～156バイトが，クラスタ0～151に対応します．各バイトは表1の意味を持っています．

ユーザーエリアは，トラック2以降に割り当てられ，0～151のクラスタがこれに対応します．

ディレクトリは，32バイトずつ，ひとつのファイルに対応します．これはディレクトリ・スロットと呼ばれます．ファイルのプロテクトの有無はディレクトリには書かれていません．ディレクトリの構成を図1に示します．

表1　ディレクトリ・スロットの構成

データ(16進)	意　味（クラスタの使用状態）
0～97	このクラスタは使用中であり，後続するクラスタを持つ．値が後続するクラスタの番号を示す．
C0～C7	このクラスタは使用中であり，ファイル最後のクラスタである．この値からC0を引いた値は，クラスタ内の実際に使用されているセクタ数＋1を表わす．
FD	クラスタ中の使用セクタはない．
FE	このクラスタはシステム領域である．
FF	このクラスタは未使用である．

使用法

プログラムをRUNさせると，ドライブ番号，トラック番号，セクタ番号をきいてくるので，処理したいドライブ，トラック，セクタ番号を入力します．その後，内容を16進数とASCIIで表示し，コマンド待ちになります．セクタ数が負の場合，内容の表示はしません．

C……セクタ内容の変更
S……そのセクタのSAVE．ドライブ，トラック，セクタ番号を指定し，書き込む．
T……クラスタ番号を16進数で入力すると，FATにおけるアドレス，トラック，セクタ番号を表示する．
P……セクタ内容のプリント・アウト
N……次に処理したいドライブ，トラック，セクタ番号の入力を要求する．番号を入力しないと，トラックはそのままで，次のセクタをLOADする．最終セクタを越えると，次のトラックに移る．
E……処理を終える．

注　意

S(セーブ)コマンドを使うときは，間違えないようにしてください．また，FATおよび処理したいファイルを含むディレクトリは，Pコマンドでプリント・アウトしてから，作業に取り組むと良いでしょう．

図1　ディレクトリ・スロットの構成

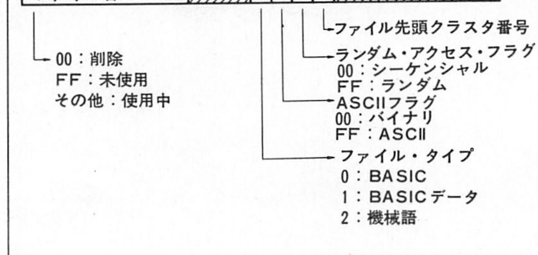


図2　FATの例

```
      Drive  number = 0
      Track  number = 1
      Secter number = 1

        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    0123456789ABCDEF

 00     00 FF FF FF FF 01 02 C2 FF FF 06 14 C0 09 0A 0B    .......ﾂ....ﾀ...
 10     C1 FF FF FF FF 11 12 C7 19 15 16 17 18 C7 1A 1B    ﾁ......ﾇ.....ﾇ..
 20     1C 1D 1E C0 C0 C0 FF FF FF FF FF FF FF FF FF FF    ...ﾀﾀﾀ..........
 30     FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
 40     FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
 50     FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
 60     FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
 70     FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
 80     FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
 90     FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
 A0     FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
 B0     FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
 C0     FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
 D0     FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
 E0     FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
 F0     FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
```

図3　ディレクトリの例

```
      Drive  number = 0
      Track  number = 1
      Secter number = 4

        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    0123456789ABCDEF

 00     53 4E 41 4B 59 20 20 20 00 00 00 00 00 00 00 00    SNAKY...........
 10     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
 20     00 20 20 20 20 20 20 20 00 00 00 00 FF 00 03 00    ................
 30     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
 40     4C 4F 41 44 52 53 20 20 00 00 00 00 00 00 07 00    LOADRS..........
 50     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
 60     50 55 43 4B 20 20 20 20 00 00 00 00 00 00 08 00    PUCK............
 70     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
 80     00 20 20 20 20 20 20 20 00 00 00 00 00 00 0C 00    ................
 90     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
 A0     41 4C 49 45 4E 20 20 20 00 00 00 00 00 00 10 00    ALIEN...........
 B0     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
 C0     32 30 30 31 20 20 20 20 00 00 00 01 FF 00 05 00    2001............
 D0     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
 E0     32 30 30 31 2D 31 20 20 00 00 00 00 00 00 13 00    2001-1..........
 F0     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
```

図4　マシン語ファイルの例

```
      Drive  number = 0
      Track  number = 16
      Secter number = 1

        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    0123456789ABCDEF

 00     00 00 36 50 00 1A 10 7F FD 07 7F FD 07 7F FD 07    ..6P............
 10     86 40 B7 FD 07 86 CE B7 FD 07 86 35 B7 FD 07 8E    ■@ｷ.■ﾎｷ.■5ｷ.■
 20     60 00 B6 FD 07 46 46 24 F9 B6 FD 06 A7 80 8C 60    `.ｶ.FF$ｶ.ｧ_■`
 30     10 26 EF 86 10 B7 FD 07 7E AF A2 FF 00 00 50 00    .&＼■.ｷ.~ｯ｢...P.
 40     1A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
 50     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
 60     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
 70     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
 80     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
 90     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
 A0     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
 B0     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
 C0     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
 D0     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
 E0     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
 F0     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
```

ディスケット・ドクター プログラム・リスト

```
10 DEFINT A-Z
20 CLEAR 1000
30 FIELD #0,128 AS A$,128 AS B$
40 IADR=PEEK(VARPTR(A$)+1)*256+PEEK(VARPTR(A$)+2)
50 WIDTH 80,25
60 LSW=1:DRV=0:TRK=1:SCT=0
70 CRLF$=CHR$(10)+CHR$(13)
80 '
90 '  Hellow Message Display
100 '
110 COLOR 7
120 PRINT
130 PRINT "*******************************"
140 PRINT "*****                     *****"
150 PRINT "*****  Disket Doctor      *****"
160 PRINT "*****                     *****"
170 PRINT "*****        81/11/17     *****"
180 PRINT "*****        T.S.D.       *****"
190 PRINT "*****                     *****"
200 PRINT "*******************************";CRLF$
210 '
220 ' Main Program Start Here
230 '
240 GOSUB850:GOSUB330
250 INPUT "Command .. (C)hange,(N)ext,(P)rint,(S)ave,(T)ranslate,(E)xit  ";CMD$
260 IF CMD$="C" THEN GOSUB 550:GOTO250
270 IF CMD$="S" THEN GOSUB 750:GOTO250
280 IF CMD$="T" THEN GOSUB 1080:GOTO250
290 IF CMD$="P" THEN GOSUB 360:GOTO 250
300 IF CMD$="N" THEN 210
310 IF CMD$="E" THEN END
320 PRINT CHR$(7);"Command error";CRLF$:GOTO 250
330 '  Dump on Screen or Printer  routine
340 IF LSW=0 THEN RETURN
350 OPEN"O",#1,"SCRN:":GOTO 380
360 OPEN"O",#1,"LPTO:":GOTO 380
370 IF CMD$="D" THEN OPEN"O",#1,"SCRN:": ELSE OPEN"O",#1,"LPTO:"
380 PRINT#1,CRLF$;"    Drive  number =";DRV;
390 PRINT#1,"    Track  number =";TRK;
400 PRINT#1,"    Secter number =";SCT;CRLF$
410 PRINT#1,"      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F";
420 PRINT#1,"      0123456789ABCDEF";CRLF$
430 FOR I=0 TO 255 STEP 16
440  D$=""
450  DUMP$=HEX$(INT(I/16))+HEX$(I MOD 16)+"   "
460  FOR J=I TO I+15
470   ID=PEEK(IADR+J):DUMP$=DUMP$+HEX$(ID/16)+HEX$(ID MOD 16)+" "
480   IF (ID<32) OR (ID=127) OR (ID=255) THEN D$=D$+"." :ELSE D$=D$+CHR$(ID)
490  NEXT J
500  PRINT#1,DUMP$;"    ";D$
510 NEXT I
520 PRINT#1,CRLF$
530 CLOSE 1
540 RETURN
550 '
560 '  Change routine
570 '
580 INPUT "Change address [0-FF] ";ANS$
590 LC=VAL("&H"+ANS$)
600 PRINT "BACK [/],NEXT [RETURN],END [.],HEX DATA [0-FF]";CRLF$
610 IF LC<0 THEN LC=0 :ELSE IF LC>255 THEN LC=255
620 PRINT HEX$(LC/16);HEX$(LC MOD 16);" : ";
630 ID=PEEK(IADR+LC)
640 PRINT HEX$(ID/16);HEX$(ID MOD 16);"  ";
650 INPUT ANS$
660 IF ANS$="/" THEN LC=LC-1:GOTO 610
670 IF ANS$="" THEN LC=LC+1:GOTO 610
680 IF ANS$="." THEN RETURN
690 ID=VAL("&H"+ANS$)
700 IF (ID<0) OR (ID>255) THEN 610
710 IF ID=0 THEN 720 :ELSE 730
720 IF ANS$="0" OR ANS$="00" THEN 730 :ELSE 610
730 POKE INT(IADR+LC),INT(ID): LC=LC+1
740 GOTO 610
750 '
760 ' Save routine
770 '
780 INPUT"Save to the same position [Y/N] ",ANS$
```

```
790 IF ANS$="Y"THEN 810
800 GOSUB 910
810 INPUT "Are you sure [Y/N] ",ANS$
820 IF ANS$<>"Y" THEN 250
830 DSKO$ DRV,TRK,SCT
840 RETURN
850 '
860 ' Load routine
870 '
880 GOSUB 910
890 DUMMY$=DSKI$(DRV,TRK,SCT)
900 RETURN
910 '
920 '  Drive,Track,Sector number input subroutine
930 COLOR 7
940 INPUT "Drive  number=";DR$:IF DR$="" THEN GOTO 960
950 DRV=VAL(DR$):IF (DRV<0) OR (DRV>2) THEN 940
960 INPUT "Track  number=";TR$:IF TR$="" THEN 980
970 TRK=VAL(TR$):IF (TRK<0) OR (TRK>39) THEN 960
980 INPUT "Sector  number=";SC$:IF SC$="" THEN SCT=SCT+1:GOTO 1030
990 SC=VAL(SC$):SCT=ABS(SC):IF (SCT<1) OR (SCT>32) THEN 980
1000 LSW=1:IF SC<0 THEN LSW=0
1010 PRINT
1020 RETURN
1030 IF SCT>32 THEN SCT=1:TRK=TRK+1
1040 RETURN
1050 '
1060 '  Traslate cluster no. to truck/sector
1070 '
1080 INPUT"Cluster number =";CL$:CL=VAL("&H"+CL$)
1090 IF (CL<0) OR (CL>156) THEN 1080
1100 PRINTCRLF$;"   FAT address=    ";HEX$(CL+5)
1110 PRINT"   Track number=  ";CL¥4+2
1120 PRINT"   Sector number= ";(CL MOD 4)*8+1;CRLF$
1130 RETURN
```

FM-8

日本語ワード・プロセッサの制作

■今久留主　洋

毎月I/Oを楽しく読ませてもらっています．実はI/Oの1981年12月号の漢字入力タブレットのプログラムをもとに，タブレットを必要としない日本語ワード・プロセッサを作ってみました．

構成図

メイン プログラム	⟶ ⟵	文字検索 プログラム

操作方法

まず，カナ・シフト・キーを押してもらいます．通常は，このモードで入力します．

書きたい漢字の音読み（たまに訓読み）のキー押すと，画面最下段に40字漢字があらわれるとともに，漢字の下にカーソルがでます．

カーソル移動キー（ → または ← ）を使って選択してもらいますが，もし，捜す漢字がない場合は，カーソル移動キー（ ↑ または ↓ ）を使って，次の40文字を画面に出してください． ↑ または ↓ を使う際，その音読みの漢字の一番端まで行くとBEEP音が出て，他の音読みの漢字まで行かないようになっています．捜す漢字が見つかれば， RETURN キーで，画面上部のカーソルの場所に表示され，なければ， SPACE キーによって，メイン・プログラムに戻ります．

キーの動き メイン・プログラムのとき

Home……………… HOME キー，
Clear screen…… CLEAR キー，
List………………… ESC キー，
Delete……………… DEL キー，
Return……………… RETURN キー，
Back Space……… BS キー，
Save………………… GRAPH キーを押しながら S キー，
Load………………… 〃 L キー，
Insert……………… 〃 I キー，
Copy………………… 〃 H キー，
カーソル移動……カーソル移動キーです．
0～9とA～Z，a～zはそのまま入力します．

文字検索プログラムの呼び出し

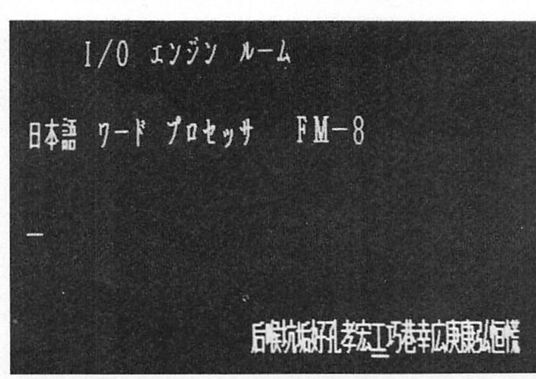
カナで漢字を表示します．
@または゛で記号，
ンまたは。でひらがな，
ー　　でカタカナ，
GRAPH キーを押しながら G でギリシア語，
〃 ロ でロシア語です．

キーの動き 文字検索プログラムのとき

← → でカーソル移動
↑ でもとの40文字表示
↓ で次の40文字表示
RETURN キーでメイン・プログラムへ，そのときのカーソル位置にある漢字データを出力します．出力したくないときは SPACE キーでメイン・プログラムに戻ります．

おわりに 長所・短所

このプログラムの長所は，タブレットなしで，漢字ROMに入っているすべての文字を文字コードを調べなくても出



力できるという点と，元のプログラムどおり，Deleteなどの Edit機能がそのまま残っているところだと思います．

このプログラムの短所は，操作方法になれないと（特に漢字の音読みがスッとでてこないと）たやすく漢字が出力できない点と，漢字コードが連続していないので，次の表示へとスクロールしていくうちに，他の音読みの漢字があらわれてくる点です．

後者は，多少複雑になるかもしれませんが，これから，頑張ってプログラムを改良していくつもりです．

なお，このプログラム作成にあたって，大阪Computer11の皆さんには大変お世話になりました．ありがとうございます．

プログラム・リスト

```
10 ' JAPANESE WORD PROCESSOR
20 DEFINT A-Z:DIM SCRN(32,8):DIM DS(60)
30 WIDTH80,25
40 FOR I=1 TO 55
50 READ A:DS(I)=A:NEXT I
60 '
70 COLOR ,7:CLS
80 SYMBOL(100,50),"Word",5,3,1
90 SYMBOL(150,100),"prosessor",5,3,1
100 SYMBOL(300,150),"I/O engine room",2,2,2:'Arranged by IMAKURUSU
110 FOR I=0 TO 2000:NEXT I:COLOR,0
120 CLS:FOR I=0 TO 32:FOR J=0 TO 8:SCRN(I,J)=0:NEXT J,I
130 I=0:J=0
140 GOSUB 450
150 A$=INKEY$:IF A$="" THEN 150 ELSE A=ASC(A$)
160 GOSUB 450
170 IF (A>27 AND A<32) THEN ON A-27 GOTO 1070,1080,1090,1100:REM ** CURSOR MOVE
180 IF A=64 OR A=222 THEN A=222:GOTO 1160:' ｷｺﾞｳ "@"or "ﾞ"
190 IF A=221 OR A=223 THEN A=224:GOTO 1160: ' ﾋﾗｶﾞﾅ"ﾝ"or"ﾟ"
200 IF A=176 THEN A=226:GOTO 1160:' ｶﾀｶﾅ"ｰ"
210 IF A=158 THEN A=228:GOTO 1160:' ｷﾞﾘｼｱｺﾞ"G"
220 IF A=224 THEN A=230:GOTO 1160:' ﾛｼｱｺﾞ"ﾛ"
230 IF A= 11 THEN 130:REM HOME
240 IF A= 12 THEN 120:REM CLEAR SCREEN
250 IF A=  5 THEN 750:REM ** ERASE LINE
260 IF A=150 THEN 540:REM ** SAVE
270 IF A=142 THEN 570:REM ** LOAD
280 IF A=232 THEN 610:REM ** INSERT
290 IF A= 27 THEN 890:REM ** LIST
300 IF A=159 THEN HARDC2:GOTO140:REM ** COPY
310 IF A=127 THEN 690:REM **DELETE
320 IF A= 13 THEN 970:REM **RETURN
330 IF A=  8 THEN 500:REM **BACK SPACE
340 IF (A>47 AND A<58) THEN D=A+8960 :GOTO 390
350 IF (A>64 AND A<91) THEN D=A+8960 :GOTO 390
360 IF (A>96 AND A<123) THEN D=A+8960 :GOTO 390
370 IF (A>176 AND A<222) THEN GOTO1160
380 GOTO 140
390 LINE(I*20,J*20)-(I*20+19,J*20+19),PRESET,,BF:PRINT@(I*20,J*20),D
400 SCRN(I,J)=D:I=I+1
410 IF I=32 THEN SCRN(32,J)=-1:I=0:J=J+1
420 IF J=9 THEN BEEP:J=8
430 GOTO 140
440 '
450 XC=I*20:YC=J*20
460 IF I=32 THEN XC=0:YC=YC+20
470 LINE(XC,YC+17)-(XC+17,YC+17),XOR,7
480 RETURN
490 '
500 I=I-1
510 IF I=-1 THEN IF J=0 THEN BEEP:GOTO 130 ELSE I=31:J=J-1:SCRN(32,J)=-1
520 GOTO 140
530 '
540 CLS:PRINT@(0,0),17744,20287:GOSUB 590
550 OPEN"O",#1,FI$:FOR J=0 TO 8:FOR I=0 TO 32:PRINT#1,SCRN(I,J):NEXT I,J:CLOSE#1
560 GOTO 880
570 CLS:PRINT@(0,0),14406,15696:GOSUB590:OPEN"I",#1,FI$
580 CLS:FOR J=0 TO 8:FOR I=0 TO 32:INPUT#1,SCRN(I,J):PRINT@(I*20,J*20),SCRN(I,J)
:NEXT I,J:CLOSE#1:GOTO 130
590 LOCATE0,3:INPUT"FILE NAME?";FI$:INPUT"DISK DRIVE?";DR:FI$=RIGHT$(STR$(DR),1)
+":"+FI$:RETURN
600 '
610 GOSUB820
620 IF NOT(SCRN(31,L-1)=0 OR SCRN(31,L-1)=8481)THEN BEEP:GOTO 140
630 FOR C=L*32-2 TO I+J*32 STEP-1:C1=C+1
640 GOSUB1000
650 NEXT C
660 SCRN(I,J)=0:LINE(I*20,J*20)-(I*20+19,J*20+19),PRESET,,BF
670 GOTO 140
680 '
```

```
690 GOSUB820
700 FOR C=I+J*32+1 TO L*32-1:C1=C-1
710 GOSUB1000
720 NEXT C
730 L=L-1:LINE(620,L*20)-(639,L*20+19),PRESET,,BF
740 GOTO 140
750 '
760 GOSUB810
770 FOR C=I+J*32 TO L*32-1
780 SCRN((C MOD 32),INT(C/32))=0
790 LINE((C MOD 32)*20,INT(C/32)*20)-((C MOD 32)*20+19,INT(C/32)*20+19),PRESET,,
BF
800 NEXT C:GOTO 140
810 '
820 L=J
830 WHILE SCRN(32,L) AND (L<8)
840 L=L+1
850 WEND
860 L=L+1
870 RETURN
880 '
890 CLS
900 FOR Y=0 TO 8
910 FOR X=0 TO31
920 PRINT@(X*20,Y*20),SCRN(X,Y)
930 NEXT X
940 NEXT Y
950 GOTO 130
960 '
970 SCRN(32,J)=0:I=0:J=J+1
980 IF J=9 THEN BEEP:J=8
990 GOTO 140
1000 '
1010 X1=C1 MOD 32:Y1=INT(C1/32):X2=C MOD 32:Y2=INT(C/32)
1020 SCRN(X1,Y1)=SCRN(X2,Y2)
1030 LINE(X1*20,Y1*20)-(X1*20+19,Y1*20+19),PRESET,,BF
1040 PRINT@(X1*20,Y1*20),SCRN(X1,Y1)
1050 RETURN
1060 '
1070 I=I+1:GOTO 1110
1080 I=I-1:GOTO 1110
1090 J=J-1:GOTO 1110
1100 J=J+1:GOTO 1110
1110 IF I=32 THEN I=0
1120 IF I=-1 THEN I=31
1130 IF J= 9 THEN J=0
1140 IF J=-1 THEN J=8
1150 GOTO 140
1160 GOSUB450:P=A-176:S=DS(P):E=DS(P+1):K=0
1170 GOSUB1360
1180 H=0:FOR M=S TO S+39
1190 PRINT@(H*16,180),M;:H=H+1:NEXT M
1200 LINE(K*16,197)-(K*16+17,197),XOR,7
1210 B$=INKEY$:IF B$="" THEN 1210
1220 IF B$=" " THEN GOSUB1360:GOTO 150 ELSE B=ASC(B$)
1230 LINE(K*16,197)-(K*16+17,197),XOR,7
1240 IF B=28 THEN K=K+1:GOTO 1300
1250 IF B=29 THEN K=K-1:GOTO 1300
1260 IF B=30 THENS=S-40:GOTO 1330
1270 IF B=31 THENS=S+40:GOTO 1330
1280 IF B=13 THEN D=S+K:GOSUB1360:GOTO 390
1290 GOTO 1210
1300 IF K=40 THEN K=0
1310 IF K=-1 THEN K=39
1320 GOTO 1200
1330 IF S<DS(P) THEN BEEP :S=S+40
1340 IF S>DS(P+1) THEN BEEP :S=S-40
1350 K=0:GOTO 1170
1360 LINE(0,180)-(639,199),PRESET,,BF:RETURN
1370 DATA &H3020,&H304A,&H3126,&H3141,&H3177
1380 DATA &H323C,&H346B,&H3665,&H3735,&H3843
1390 DATA &H3A33,&H3B45,&H3F5A,&H4024,&H4139
1400 DATA &H423E,&H434D,&H4445,&H4462,&H4546
1410 DATA &H4660,&H4673,&H4720,&H4729,&H4735
1420 DATA &H4743,&H485B,&H4954,&H4A3A,&H4A5D
1430 DATA &H4B60,&H4C23,&H4C33,&H4C3D,&H4C4E
1440 DATA &H4C69,&H4C7B,&H4D3D,&H4D65,&H4D78
1450 DATA &H4E5C,&H4E61,&H4F24,&H4F41,&H4F54
1460 DATA &H2122,&H222F,&H2421,&H2474,&H2521,&H2576
1470 DATA &H2621,&H2659,&H2721,&H2772
```

ゲーム論



FM-8活用研究



■高畑英樹

いままでI/O誌に発表されているゲーム・ソフトは，ゲームセンターなどでよく見かけるゲーム内容のものがほとんどで，やはりそれにはマシン語で記述されているプログラムでなければ，スピードの点でメリットがないと思います．しかし，FM-8では，アセンブラなどが使えるのは4～5月ごろとのことで，それまでに何とかBASICでゲームらしき物をと考えて作ってみました．

ゲームの発想

色々な命令（グラフィック処理）を操作しているうちに，1つの命令を実行するのは非常に速く，LINE，CONNECT，CIRCLEなどに至ってはマシン語並の速さで処理していることがわかりました．

これはBASICが命令を解読した後，マシン語レベルで処理していると思われ，一度開始すると，割り込みキーもストップ・キーも無視して1つの命令に専念しているようです．

また，変数などを介入させたGET@，PUT@文については，同一内容の処理をするCONNECTとPAINT文より，スピードが遅いようです．

そこで今回はGET@，PUT@文はまったく使わずCONNECT，PAINT文でグラフィック・パターンを処理しています．

そして何回かライン命令を実行しているうちに，これが，レーザービームに見えてくるではありませんか．そしてサークル命令が，これを防ぐバリヤに…….

もう小道具は揃ったようです．そこで空中戦でもと思い，さっそく着手しました．しかし，表示がいかに速くても，それらを組み立てる手段に時間を費いやしたのでは，スピードのあるゲームは作れません．

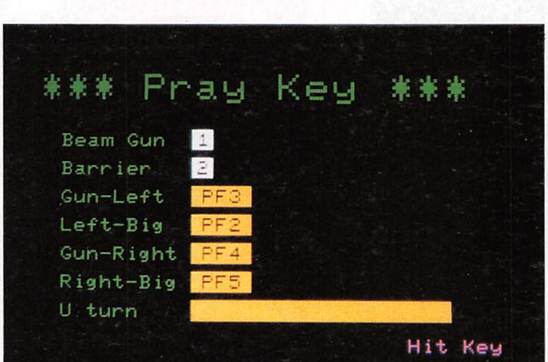
そこで，多少リストは見にくくなりますが，マルチステートメントの多用や同一内容の処理でもサブルーチン化せず，ダイレクトに処理をする方法を取りました．リストを見ると同じようなプログラムがだらだらと長く続いていますが，これも高スピード化のためです．

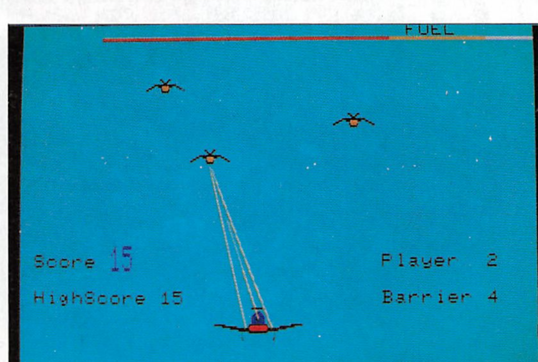
全体の表示はプログラム・スイッチにより切り替えて順次表示しているので，スムーズな動きをします．要するに，『1つの処理を最短コースで』を念頭に作りました．

BASICプログラムでは，全体的に，はでな動きは表現できませんが，敵戦闘機の表示は遠近法により，同一パターンの拡大・縮少で表現し臨場感を得て，しかも接近時間を自由に設定できて，迫力のあるゲームとなりました．プレイヤーのビーム・ガンの方向操作は全体の同一ループ内で処理をしていたのでは，急速な方向決定は望めないため，ファンクション・キーを使い，メイン・ルーチンに割り込みをかけて処理をするようにしました．全体の処理経過は，インターバル割り込みで処理しています．

最終的に，このゲームをしていて，これが，BASICで書かれたプログラムかと，疑いたくなるほどスピーディーな表現を見せてくれました．

ゲーム内容の説明





まず，敵戦闘機は常に3機表示され，常に画面手前側へ飛行を続けており，引き帰すことは一切しません．そのスピードは6段階で接近してきます．ただし，1分経過後ごとに3機のうち1機が倍速(3段階表示)で接近し，3分後にすべての敵機が倍速になります．

そして攻撃は，最大接近（敵機の色が黄色になる）のときに1クッションたって，ビームを発射します．そしてこのビーム銃の機能は，到達距離は短かい代りに，一度発射すると命中率が100％で一発でプレイヤー機を破壊します．

したがって，敵機の攻撃エリアに入る前にプレイヤー機はUターンして敵機との距離をとるか，またはUターン中に攻撃される危険のあるときは，バリアを張ってUターンすると安全です．そして敵機との距離が開くと，再びUターンして対戦します．

次にプレイヤー機のレーザービーム・ガン機能は到達距離が抜群に長い代り，照準を敵機に合わなければ破壊できません．照準の合わせ方は**PF2**～**PF5**のファンクション・キーを使って左右に移動させます．

PF2と**PF5**はそれぞれ左右に3ポイント一度に移動します．照準点の現在位置は，FUELインジケータの下側にピンクの＋印で表示しています．

バリアは初期設定が3で10機破壊するごとに1つ追加されます．4分15秒で時間切れのゲーム・オーバーになります．もちろん，その前にプレイヤー機が3機破壊されるとゲーム・オーバーになります．

その他

敵機の破壊音がときどき乱れることがありますが，これは8秒ごとにタイマー割り込みがあることと，ファンクション・キー割り込みがあり，プログラムの中の**BEEP1**と**BEEP0**のループ処理の最中に割り込みが発生すると起こります．しかし，割り込み処理の内容が短いため，そんなに気にすることはありません．

また，給油後の**3**キーインの提示は，1つ前の場面で**1**，**2**，**スペース**キーが押され，それが実行されずに次の局面に移り再開されたとき，一度もキーインしないのに前回押されたキー処理がなされてしまったためです．

この対策はゲーム再開のときKEY＄の解除の命令があれば**3**キーインの指示は不要になりますが，KEY＄に0を代入したりして色々と試みましたが，キー入力の解除はできませんでした．

ハード的なものかまたはソフトでできるものか見当がつかず，読者皆さんで直接POKE文などを使ってキーバッファかフラグ類をキャンセルできないものか教えてください．

また，今回はプログラマブル・ファンクション・キーを使っていますが，これも連続キーが本体のキーで使えると，使わなくて済んだのです．

本体キーはキー押し下げ直後のキー入力の確認は即実行しますが，同一キーの押し下げが連続しているにもかかわらず1クッションの間をおいて，後に連続キーの入力確認をしているため，ゲーム進行上この1クッションは非情にも致命的となります．

これについてもキーフラグ，バッファなどをPOKE文で直接キャンセルして，このアクションを防げないものでしょうか．

ハード的にできないときはあきらめて，ジョイスティックを使うことになりますが，これは一般的ではありません．これについても皆さんからの発表を期待します．

注意

タイマーが常に割り込みをかけて処理をしていますから，たとえばカセットのセーブ，ロードのタイミングが乱され，セーブ，ロードが不能になります．

この場合マニュアルに書かれているように，インターバルOFFとキー（2～5）OFFを実行してください．ただし，プログラムをゲーム・オーバーの表示のあるまで走らせた場合は，自動的にOFFの処理をします．プレイヤーが1回負けるごとにタイマーをSTOPしていますが，OFFにはしていません．また，ファンクション・キーも無効のままです．

プログラムの説明

全体のフローチャートとしては割り込みが終了し，かつキー入力のないとき常にループしている文は，330～370の5つの文だけです．

ただし，330の文は1170実行後，330の**IF**文に帰り，そのときプレイヤーを攻撃しているとフラグTを1にして帰り，1080ジャンプします（1170は敵機の表示サブルーチン，1080はプレイヤー爆発サブルーチンです）．

370の**IF**文はバリアを一定時間経過後消去するルーチンです．その他走らないときはFM-8のエラー機能が充実しているので，個々に検討してください．

それでは，この素晴らしいスピードと迫力のあるゲームをさっそく入力して楽しんでください．

FIGHTER プログラム・リスト

```
10 WIDTH40,20:COLOR4,0:CLS:SYMBOL(90,70),"ｶﾝｼﾞ ﾋｰｶﾝｼﾞROM",4,2,4:SYMBOL(100,100),
"ｼﾞﾂｿｳ ｼﾃｲﾙ ﾄｷ Y ｦｵｽ",2,1:SYMBOL(100,120),"ｼﾞﾂｿｳ ｼﾃｲﾅｲ ﾄｷ N ｦｵｽ",2,1:A$=INPUT$(1
):IFA$="N"ORA$="n"THEN70
20 CLS:PRINT@(230,38),&H4250,&H3020,&H406F,&H3020,&H244E,&H3020,&H4062,&H3020,&H
4C40:PRINT@(100,68),&H4528,&H406F,&H462E,&H3521,&H244F:LINE(185,69)-(245,83),PSE
T,5,BF:X=190:COLOR0:GOSUB980:COLOR4:PRINT@(250,68),&H244E,&H3741,&H244E,&H3B7E,&
H3936,&H3762
30 PRINT@(348,68),&H2472,&H332B,&H3B4F,&H2439,&H246B,&H2123:PRINT@(100,88),&H326
6,&H406F,&H462E,&H3521,&H244F,&H4C5C,&H4130,&H2447,&H4528,&H3521,&H2472,&H334E,&
H4727,&H3C21,&H4268,&H2439,&H2459,&H2446,&H3936,&H3762,&H2447,&H242D,&H246B,&H21
23
40 PRINT@(50,118),&H4250,&H406F,&H244F,&H416A,&H4250,&H387E,&H2437,&H2446,&H3954
,&H244A,&H2426,&H243F,&H2461,&H405C,&H3661,&H4130,&H244B,&H2122,&H2439,&H2459,&H
2446,&H2472,&H474B,&H3275,&H2439,&H246B,&H242B,&H4B74,&H244F,&H405C,&H3661,&H243
5,&H246C,&H2446
```

FIGHTER プログラム・リスト

```
50 PRINT@(50,138),&H4528,&H3521,&H244E,&H3936,&H3762,&H484F,&H304F,&H244B,&H467E
,&H2443,&H243F,&H3649,&H4C4C,&H2447,&H244F,&H4545,&H3C27,&H2550,&H256A,&H2564,&H
2472,&H4425,&H246B,&H242B,&H2122,&H307A,&H242D,&H4A56,&H2428,&H2437,&H2446,&H212
2,&H4528,&H244B
60 PRINT@(50,158),&H244E,&H3577,&H4E25,&H2472,&H4640,&H2446,&H3A46,&H2453,&H4250
,&H387E,&H2437,&H2446,&H4250,&H406F,&H2439,&H246B,&H2123:SYMBOL(525,178),"Hit Ke
y",1,1,3:A$=INPUT$(1):GOTO100
70 COLOR4,0:CLS:SYMBOL(150,50),"*** ﾀｲｾﾝ ﾉ ｾﾂﾒｲ ***",2,1:SYMBOL(15,72),"ﾃｷ ﾉ ｾﾝﾄ
ｳｷ ﾊ     ｿ ﾊﾟﾀｰﾝ ﾉ ﾄｷﾆ ｺｳｹﾞｷ ｽﾙ",2,1:X=215:LINE(210,71)-(270,80),PSET,5,BF:COLOR0
:GOSUB980:COLOR4:SYMBOL(15,83),"ﾜｶﾞ ｾﾝﾄｳｷﾊ ﾃｷｷ ｶｸﾆﾝ ｼﾀﾞｲ ｽﾍﾞﾃｺｳｹﾞｷﾃﾞｷﾙ",2,1
80 LOCATE4,11:PRINT"ﾀｲｾﾝ ﾊ ﾃｷｷ ﾄ ｱｲ ﾀｲｺｳ ｼﾃ ｵｺﾅｳ ﾀﾒ";:LOCATE3,12:PRINT"ｾﾂｷﾝ ﾏｴﾆ
ｽﾍﾞﾃ ｵ ﾊｶｲ ｽﾙｶ ﾏﾀﾜ ｾﾂｷﾝ";:LOCATE3,13:PRINT"ｻﾚﾃ ﾃｷｷ ﾉ ｺｳｹﾞｷ ｴﾘﾔﾆ ﾊｲﾂﾀ ｷｮｸﾒﾝ";:LOC
ATE3,14:PRINT"ﾃﾞﾜ ﾃﾞﾝｼﾞﾊﾞﾘﾔ ｦ ﾊﾙｶ ﾋｷｶｴｼﾃ ﾃｷ ﾄﾉ";:LOCATE3,15:PRINT"ｷｮﾘ ｦ ｴﾃ ﾌﾀﾀﾋﾞ
 ﾀｲｺｳｼﾃ ﾀｲｾﾝｽﾙ"
90 SYMBOL(500,180),"Hit Key",2,1,3:A$=INPUT$(1):GOTO100
100 CLS:SYMBOL(60,40),"*** Pray Key ***",4,2:LINE(250,70)-(275,80),PSET,7,BF:LIN
E(250,85)-(275,95),PSET,7,BF:X=100:FORI=1TO4:LINE(250,X)-(320,X+10),PSET,6,BF:X=
X+15:NEXT:LINE(250,160)-(550,170),PSET,6,BF:SYMBOL(255,72),"1",2,1,0:SYMBOL(255,
87),"2",2,1,0
110 SYMBOL(262,102),"PF3",2,1,0:SYMBOL(262,117),"PF2",2,1,0:SYMBOL(262,132),"PF4
",2,1,0:SYMBOL(262,147),"PF5",2,1,0:SYMBOL(90,72),"Beam Gun",2,1,4:SYMBOL(90,87)
,"Barrier",2,1,4:SYMBOL(90,102),"Gun-Left",2,1,4:SYMBOL(90,117),"Left-Big",2,1,4
120 SYMBOL(90,132),"Gun-Right",2,1,4:SYMBOL(90,147),"Right-Big",2,1,4:SYMBOL(90,
162),"U turn",2,1,4:SYMBOL(500,180),"Hit Key",2,1,3:A$=INPUT$(1)
130 DIMH%(0):DIMP(60):HS=0:FORI=1TO60:READP(I):NEXT:GOTO1460
140 LINE(15,15)-(22,17)," ",5,BF:COLOR0
150 CONNECT(300,140)-(300,143)-(302,144)-(302,148)-(308,150)-(308,155)-(354,165)
-(354,162)-(356,162)-(356,170)-(354,170)-(354,167)-(308,161)-(308,170)-(312,173)
-(288,173)-(292,170)-(292,161)-(246,167)-(246,170)-(244,170)-(244,162)-(246,162)
-(246,165),0
160  CONNECT(246,165)-(292,155)-(292,150)-(298,148)-(298,144)-(300,143),:CONNECT
(296,160)-(296,153)-(298,152)-(302,152)-(304,153)-(304,160),:CONNECT(300,169)-(3
00,173),:PAINT(301,172),1,0:FORU=0TO400:NEXT:GOSUB1170
170 IFT=1THEN1080ELSELINE(15,14)-(22,17)," ",5,BF
180 CONNECT(284,165)-(282,166)-(280,166)-(284,187)-(278,167)-(278,166)-(276,165)
-(276,155)-(278,154)-(278,153)-(284,133)-(280,154)-(282,154)-(284,155),0:CONNECT
(282,158)-(286,156)-(292,156)-(298,159)-(298,161)-(292,164)-(286,164)-(282,162)-
(282,158),1
190 CONNECT(280,150)-(272,150)-(268,151),0:CONNECT(280,170)-(272,170)-(268,169),
:CONNECT(302,156)-(302,164),:CONNECT(299,160)-(302,160),:CONNECT(290,159)-(292,1
60)-(290,161)-(288,161)-(286,160)-(288,159)-(290,159),3:PAINT(294,160),7,1,3:PAI
NT(290,160),7,3
200 PAINT(280,160),1,0,1:FORU=0TO300:NEXT:GOSUB1170:IFT=1THENGOTO1080:ELSELINE(1
6,13)-(18,18)," ",5,BF
210 CONNECT(292,168)-(288,169)-(288,170)-(246,168)-(286,171)-(288,171)-(290,172)
-(310,172)-(312,171)-(314,171)-(354,168)-(312,170)-(312,169)-(308,168),0:CONNECT
(280,170)-(280,175)-(282,176),:CONNECT(320,170)-(320,175)-(318,176),:CONNECT(292
,159)-(308,159)
220 PSET(300,160):CONNECT(300,161)-(302,161)-(308,164)-(308,167)-(304,169)-(296,
169)-(292,167)-(292,164)-(298,161)-(300,161),1:CONNECT(297,165)-(300,164)-(303,1
65)-(303,166)-(300,167)-(297,166)-(297,165),3:PAINT(300,163),7,1,3:PAINT(300,165
),7,3
230 PAINT(300,171),1,0,1:GOSUB1170:F=0:GOTO330
240 F=1:LINE(15,15)-(22,17)," ",5,BF:COLOR0
250 CONNECT(300,140)-(300,143)-(302,144)-(302,148)-(308,150)-(308,155)-(354,165)
-(354,162)-(356,162)-(356,170)-(354,170)-(354,167)-(308,161)-(308,170)-(312,173)
-(288,173)-(292,170)-(292,161)-(246,167)-(246,170)-(244,170)-(244,162)-(246,162)
-(246,165),0
260  CONNECT(246,165)-(292,155)-(292,150)-(298,148)-(298,144)-(300,143),:CONNECT
(282,160)-(284,160)-(284,158),7:CONNECT(318,160)-(316,160)-(316,158),7:PAINT(300
,172),0,0,7:FORU=0TO400:NEXT:GOSUB1170:IFT=1THEN1080:ELSELINE(15,14)-(22,17)," "
,5,BF
270 CONNECT(316,133)-(322,153)-(322,154)-(324,155)-(324,165)-(322,166)-(322,167)
-(316,187)-(320,166)-(318,166)-(316,165)-(316,155)-(318,154)-(320,154)-(316,133)
,0:CONNECT(316,164)-(308,164)-(302,161)-(302,159)-(308,156)-(316,156),1
280 CONNECT(298,156)-(298,164),0:CONNECT(298,160)-(316,160)-(316,161)-(310,161)-
(310,159)-(316,159),:CONNECT(320,150)-(328,150)-(332,151),:CONNECT(320,170)-(328
,170)-(332,169),:PAINT(310,158),1,0,1:PAINT(308,162),1,0,1:PAINT(320,160),C,0,1
290 FORU=0TO300:NEXT:GOSUB1170:IFT=1THEN1080:ELSELINE(18,13)-(20,18)," ",5,BF
300 CONNECT(290,168)-(288,169)-(288,170)-(246,168)-(286,171)-(288,171)-(290,172)
-(310,172)-(312,171)-(314,171)-(354,168)-(312,170)-(312,169)-(310,168),0:CONNECT
(280,170)-(280,175)-(282,176),:CONNECT(320,170)-(320,175)-(318,176),:CONNECT(292
,159)-(308,159)
310 LINE(290,168)-(310,168),PSET:CONNECT(292,167)-(292,164)-(298,161)-(302,161)-
(308,164)-(308,167),1:PAINT(300,165),1,0,1:PAINT(300,171),C,0:LINE(300,159)-(300
,168),PSET,0:CONNECT(298,167)-(298,165)-(302,165)-(302,167),0:GOTO330
```

```
320 DATA16,270,20,328,13,214,24,386,9,158,27,444,6,100,31,500,2,44,34,556,18,296
,15,250,21,344,12,200,24,392,9,154,27,440,6,106,30,488,3,58,15,254,18,292,19,310
,20,330,14,234,16,272,21,350,13,216,6,98,29,468
330 GOSUB1170:IFT=1THEN1080:ELSEF$=INKEY$
340 IFF$=" "THEN480
350 IFF$="1"ANDF=1THENI=G:D=Z:LINE(300,158)-(I,20),PSET,2:LINE(300,158)-(I,20),P
RESET:ON D GOTO510,530,540,550,560,570,580,600,620,640,660,680,700,720,740,750,7
60,770,780,790
360 IFF$="2"ANDF=1ANDB1-1>0THENCIRCLE(300,150),120,4,.2,.5,0,N:B=1:B1=B1-1:Q=0:L
OCATE35,15:COLOR0:PRINTB1-1:GOTO140
370 IFB=1THENQ=Q+1:IFQ=5THENCIRCLE(300,150),120,5,.2,.5,0,N:B=0:GOTO330
380 GOTO330
390 INTERVALOFF:SYMBOL(100,90),"GAME OVER !!",5,2,2:SYMBOL(10,190),"ｹﾞｰﾑ ｦ ｽﾙﾄｷ
ﾜ Y ﾔﾒﾙﾄｷ ﾜ N ｦ ｵｽ",2,1,0:FORI=0TO3000:NEXT:KEY(2)OFF:KEY(3)OFF:KEY(4)OFF:KEY(5)
OFF
400 A$=INKEY$:IFA$="N"ORA$="n"THENCOLOR4,0:CLS:SYMBOL(50,90),"ｵﾂｶﾚ ｻﾏ ﾃﾞｼﾀ !",5,
2,4:SYMBOL(500,180),"KEY IN Y OR STOP",1,1:ELSEIFA$="Y"ORA$="y"THENGOTO1460
410 GOTO400
420 IFG<616THENSYMBOL(G,10),"+",1,1,5:G=G+32:Z=Z+1:SYMBOL(G,10),"+",1,1,3:RETURN
:ELSERETURN
430 IFG>8THENSYMBOL(G,10),"+",1,1,5:G=G-32:Z=Z-1:SYMBOL(G,10),"+",1,1,3:RETURN:E
LSERETURN
440 SYMBOL(G,10),"+",1,1,5:G=G+96:Z=Z+3:IFZ>20THENG=616:Z=20
450 SYMBOL(G,10),"+",1,1,3:RETURN
460 SYMBOL(G,10),"+",1,1,5:G=G-96:Z=Z-3:IFZ<1THENG=8:Z=1
470 SYMBOL(G,10),"+",1,1,3:RETURN
480 IFF=1THEN140
490 IFF=0THEN240
500 GOTO330
510 IFP3=57ANDF3>0THENK=P(P3+1):K1=72:K2=F3:F3=0:P3=59:GOSUB1100
520 IFP2=39ANDF2>0THENK=P(P2+1):K1=52:K2=F2:F2=0:P2=21:GOSUB1100:GOTO330:ELSE330
530 IFP1=17ANDF1>0THENK=P(P1+1):K1=32:K2=F1:F1=0:P1=19:GOSUB1100:GOTO330:ELSE330
540 IFP2=35ANDF2>0THENK=P(P2+1):K1=52:K2=F2:F2=0:P2=37:GOSUB1100:GOTO330:ELSE330
550 IFP1=13ANDF1>0THENK=P(P1+1):K1=32:K2=F1:F1=0:P1=15:GOSUB1100:GOTO330:ELSE330
560 IFP2=31ANDF2>0THENK=P(P2+1):K1=52:K2=F2:F2=0:P2=33:GOSUB1100:GOTO330:ELSE330
570 IFP1=9ANDF1>0THENK=P(P1+1):K1=32:K2=F1:F1=0:P1=11:GOSUB1100:GOTO330:ELSE330
580 IFP3=55ANDF3>0THENK=P(P3+1):K1=72:K2=F3:F3=0:P3=57:GOSUB1100
590 IFP2=27ANDF2>0THENK=P(P2+1):K1=52:K2=F2:F2=0:P2=29:GOSUB1100:GOTO330:ELSE330
600 IFP3=49ANDF3>0THENK=P(P3+1):K1=72:K2=F3:F3=0:P3=51:GOSUB1100
610 IFP1=5ANDF1>0THENK=P(P1+1):K1=32:K2=F1:F1=0:P1=7:GOSUB1100:GOTO330:ELSE330
620 IFP3=41ANDF3>0THENK=P(P3+1):K1=72:K2=F3:F3=0:P3=43:GOSUB1100
630 IFP2=23ANDF2>0THENK=P(P2+1):K1=52:K2=F2:F2=0:P2=25:GOSUB1100:GOTO330:ELSE330
640 IFP3=51ANDF3>0THENK=P(P3+1):K1=72:K2=F3:F3=0:P3=53:GOSUB1100
650 IFP1=1ANDF1>0THENK=P(P1+1):K1=32:K2=F1:F1=0:P1=3:GOSUB1100:GOTO330:ELSE330
660 IFP3=43ANDF3>0THENK=P(P3+1):K1=72:K2=F3:F3=0:P3=45:GOSUB1100
670 IFP2=21ANDF2>0THENK=P(P2+1):K1=52:K2=F2:F2=0:P2=23:GOSUB1100:GOTO330:ELSE330
680 IFP3=45ANDF3>0THENK=P(P3+1):K1=72:K2=F3:F3=0:P3=47:GOSUB1100
690 IFP1=3ANDF1>0THENK=P(P1+1):K1=32:K2=F1:F1=0:P1=5:GOSUB1100:GOTO330:ELSE330
700 IFP3=47ANDF3>0THENK=P(P3+1):K1=72:K2=F3:F3=0:P3=49:GOSUB1100
710 IFP2=25ANDF2>0THENK=P(P2+1):K1=52:K2=F2:F2=0:P2=27:GOSUB1100:GOTO330:ELSE330
720 IFP3=53ANDF3>0THENK=P(P3+1):K1=72:K2=F3:F3=0:P3=55:GOSUB1100
730 IFP1=7ANDF1>0THENK=P(P1+1):K1=32:K2=F1:F1=0:P1=9:GOSUB1100:GOTO330:ELSE330
740 IFP2=29ANDF2>0THENK=P(P2+1):K1=52:K2=F2:F2=0:P2=31:GOSUB1100:GOTO330:ELSE330
750 IFP1=11ANDF1>0THENK=P(P1+1):K1=32:K2=F1:F1=0:P1=13:GOSUB1100:GOTO330:ELSE330
760 IFP2=33ANDF2>0THENK=P(P2+1):K1=52:K2=F2:F2=0:P2=35:GOSUB1100:GOTO330:ELSE330
770 IFP1=15ANDF1>0THENK=P(P1+1):K1=32:K2=F1:F1=0:P1=17:GOSUB1100:GOTO330:ELSE330
780 IFP2=37ANDF2>0THENK=P(P2+1):K1=52:K2=F2:F2=0:P2=39:GOSUB1100:GOTO330:ELSE330
790 IFP3=59ANDF3>0THENK=P(P3+1):K1=72:K2=F3:F3=0:P3=41:GOSUB1100
800 IFP1=19ANDF1>0THENK=P(P1+1):K1=32:K2=F1:F1=0:P1=1:GOSUB1100:GOTO330:ELSE330
810 CONNECT(X+22,35)-(X+24,34)-(X+26,34)-(X+28,35),0:RETURN
820 CONNECT(X+22,55)-(X+24,54)-(X+26,54)-(X+28,55),0:RETURN
830 CONNECT(X+22,75)-(X+24,74)-(X+26,74)-(X+28,75),0:RETURN
840 CONNECT(X+18,36)-(X+22,35)-(X+28,35)-(X+32,36),0:CONNECT(X+24,34)-(X+24,37)-
(X+26,37)-(X+26,34),:RETURN
850 CONNECT(X+18,56)-(X+22,55)-(X+28,55)-(X+32,56),0:CONNECT(X+24,54)-(X+24,57)-
(X+26,57)-(X+26,54),:RETURN
860 CONNECT(X+18,76)-(X+22,75)-(X+28,75)-(X+32,76),0:CONNECT(X+24,74)-(X+24,77)-
(X+26,77)-(X+26,74),:RETURN
870 CONNECT(X+14,37)-(X+20,35)-(X+30,35)-(X+36,37),0:CONNECT(X+22,33)-(X+24,34)-
(X+26,34)-(X+28,33),:CONNECT(X+22,36)-(X+24,37)-(X+26,37)-(X+26,36)-(X+28,36),:R
ETURN
880 CONNECT(X+14,57)-(X+20,55)-(X+30,55)-(X+36,57),0:CONNECT(X+22,53)-(X+24,54)-
(X+26,54)-(X+28,53),:CONNECT(X+22,56)-(X+24,57)-(X+26,57)-(X+26,56)-(X+28,56),:R
ETURN
890 CONNECT(X+14,77)-(X+20,75)-(X+30,75)-(X+36,77),0:CONNECT(X+22,73)-(X+24,74)-
(X+26,74)-(X+28,73),:CONNECT(X+22,76)-(X+24,77)-(X+26,77)-(X+26,76)-(X+28,76),:R
```

FIGHTER プログラム・リスト

```
ETURN
900 CONNECT(X+8,37)-(X+18,35)-(X+32,35)-(X+42,37),0:CONNECT(X+20,33)-(X+22,33)-(
X+24,34)-(X+26,34)-(X+28,33)-(X+30,33),:CONNECT(X+20,36)-(X+22,36)-(X+22,37)-(X+
24,38)-(X+26,38)-(X+28,37)-(X+28,36)-(X+30,36),:PAINT(X+26,37),3,0:RETURN
910 CONNECT(X+8,57)-(X+18,55)-(X+32,55)-(X+42,57),0:CONNECT(X+20,53)-(X+22,53)-(
X+24,54)-(X+26,54)-(X+28,53)-(X+30,53),:CONNECT(X+20,56)-(X+22,56)-(X+22,57)-(X+
24,58)-(X+26,58)-(X+28,57)-(X+28,56)-(X+30,56),:PAINT(X+26,57),3,0:RETURN
920 CONNECT(X+8,77)-(X+18,75)-(X+32,75)-(X+42,77),0:CONNECT(X+20,73)-(X+22,73)-(
X+24,74)-(X+26,74)-(X+28,73)-(X+30,73),:CONNECT(X+20,76)-(X+22,76)-(X+22,77)-(X+
24,78)-(X+26,78)-(X+28,77)-(X+28,76)-(X+30,76),:PAINT(X+26,77),3,0:RETURN
930 CONNECT(X+4,37)-(X+16,35)-(X+34,35)-(X+46,37),0:CONNECT(X+18,32)-(X+20,32)-(
X+24,34)-(X+26,34)-(X+28,32)-(X+30,32),:CONNECT(X+18,36)-(X+22,36)-(X+20,37)-(X+
20,38)-(X+22,39)-(X+28,39)-(X+30,38)-(X+30,37)-(X+32,36)-(X+28,36),:PAINT(X+24,3
7),3,0:RETURN
940 CONNECT(X+4,57)-(X+16,55)-(X+34,55)-(X+46,57),0:CONNECT(X+18,52)-(X+20,52)-(
X+24,54)-(X+26,54)-(X+28,52)-(X+30,52),:CONNECT(X+18,56)-(X+22,56)-(X+20,57)-(X+
20,58)-(X+22,59)-(X+28,59)-(X+30,58)-(X+30,57)-(X+32,56)-(X+28,56),:PAINT(X+24,5
7),3,0:RETURN
950 CONNECT(X+4,77)-(X+16,75)-(X+34,75)-(X+46,77),0:CONNECT(X+18,72)-(X+20,72)-(
X+24,74)-(X+26,74)-(X+28,72)-(X+30,72),:CONNECT(X+18,76)-(X+22,76)-(X+20,77)-(X+
20,78)-(X+22,79)-(X+28,79)-(X+30,78)-(X+30,77)-(X+32,76)-(X+28,76),:PAINT(X+24,7
7),3,0:RETURN
960 CONNECT(X,38)-(X+14,35)-(X+36,35)-(X+50,38),0:CONNECT(X+18,32)-(X+20,32)-(X+
24,34)-(X+26,34)-(X+30,32)-(X+32,32),:CONNECT(X+16,36)-(X+20,36)-(X+20,38)-(X+22
,39)-(X+28,39)-(X+30,38)-(X+30,36)-(X+34,36),:PAINT(X+24,37),6,0:RETURN
970 CONNECT(X,58)-(X+14,55)-(X+36,55)-(X+50,58),0:CONNECT(X+18,52)-(X+20,52)-(X+
24,54)-(X+26,54)-(X+30,52)-(X+32,52),:CONNECT(X+16,56)-(X+20,56)-(X+20,58)-(X+22
,59)-(X+28,59)-(X+30,58)-(X+30,56)-(X+34,56),:PAINT(X+24,57),6,0:RETURN
980 CONNECT(X,78)-(X+14,75)-(X+36,75)-(X+50,78),0:CONNECT(X+18,72)-(X+20,72)-(X+
24,74)-(X+26,74)-(X+30,72)-(X+32,72),:CONNECT(X+16,76)-(X+20,76)-(X+20,78)-(X+22
,79)-(X+28,79)-(X+30,78)-(X+30,76)-(X+34,76),:PAINT(X+24,77),6,0:RETURN
990 GOSUB960:I=P(P1+1)+24:SYMBOL(I-12,40),"*",3,1,2:BEEP1:FOR II=0TO30:NEXT:BEEP
0:SYMBOL(I-12,40),"*",3,1,5
1000 IFB=0THENLINE(I,40)-(300,165),PSET,6:LINE(I,40)-(284,170),PSET,6:LINE(I,40)
-(316,170),PSET,6:LINE(I,40)-(300,165),PRESET:LINE(I,40)-(284,170),PRESET:LINE(I
,40)-(316,170),PRESET:T=1:RETURN
1010 F1=F1-2:RETURN
1020 GOSUB970:I=P(P2+1)+24:SYMBOL(I-12,60),"*",3,1,2:BEEP1:FOR II=0TO30:NEXT:BEE
P0:SYMBOL(I-12,60),"*",3,1,5
1030 IFB=0THENLINE(I,60)-(300,165),PSET,6:LINE(I,60)-(284,170),PSET,6:LINE(I,60)
-(316,170),PSET,6:LINE(I,60)-(300,165),PRESET:LINE(I,60)-(284,170),PRESET:LINE(I
,60)-(316,170),PRESET:T=1:RETURN
1040 F2=F2-2:RETURN
1050 GOSUB980:I=P(P3+1)+24:SYMBOL(I-12,80),"*",3,1,2:BEEP1:FOR II=0TO30:NEXT:BEE
P0:SYMBOL(I-12,80),"*",3,1,5
1060 IFB=0THENLINE(I,80)-(300,165),PSET,6:LINE(I,80)-(284,170),PSET,6:LINE(I,80)
-(316,170),PSET,6:LINE(I,80)-(300,165),PRESET:LINE(I,80)-(284,170),PRESET:LINE(I
,80)-(316,170),PRESET:T=1:RETURN
1070 F3=F3-2:RETURN
1080 INTERVALSTOP:BEEP1:SYMBOL(255,155),"*",10,2,2:FORI=0TO700:NEXT:BEEP0:PL=PL-
1:LOCATE35,13:COLOR0:PRINTPL;:LINE(15,13)-(22,18)," ",5,BF:LINE(15,15)-(22,17),"
*",3,BF:LINE(15,13)-(22,17)," ",5,BF:LINE(17,16)-(20,17),"*",3,BF
1090 LINE(17,13)-(20,17)," ",5,BF:LINE(18,16)-(19,16),"*",3,BF:LINE(18,13)-(19,1
6)," ",5,BF:IFPL=0THEN390:ELSEFORI=0TO2000:NEXT:B=0:C=5:F1=0:F2=0:F3=0:G=296:Z=1
0:T=0:CLS:GOTO1330
1100 ON K2 GOTO1110,1110,1120,1120,1130,1130,1130,1130,1130
1110 SYMBOL(K+18,K1+1),"**",1,1,2:FORI=0TO20:BEEP1:BEEP0:NEXT:PO=VAL(PO$):PO=PO+
1:PO$=STR$(PO):LINE(95,125)-(190,140),PSET,5,BF:SYMBOL(95,125),PO$,2,2,1:LINE(K,
K1)-(K+50,K1+7),PSET,5,BF:IFRIGHT$(PO$,1)="0"THENGOTO1140:ELSEGOTO1150
1120 SYMBOL(K+10,K1+1),"**",2,1,2:FORI=0TO20:BEEP1:BEEP0:NEXT:PO=VAL(PO$):PO=PO+
1:PO$=STR$(PO):LINE(95,125)-(190,140),PSET,5,BF:SYMBOL(95,125),PO$,2,2,1:LINE(K,
K1)-(K+50,K1+7),PSET,5,BF:IFRIGHT$(PO$,1)="0"THENGOTO1140:ELSEGOTO1150
1130 SYMBOL(K,K1+1),"**",3,1,2:FORI=0TO20:BEEP1:BEEP0:NEXT:PO=VAL(PO$):PO=PO+1:P
O$=STR$(PO):LINE(95,125)-(190,140),PSET,5,BF:SYMBOL(95,125),PO$,2,2,1:LINE(K,K1)
-(K+50,K1+7),PSET,5,BF:IFRIGHT$(PO$,1)="0"THENGOTO1140:ELSEGOTO1150
1140 B1=B1+1:LOCATE35,15:COLOR0:PRINTB1-1
1150 IF PO>HS THEN HS=PO:LOCATE10,15:COLOR0:PRINTHS:RETURN:ELSERETURN
1160 RETURN
1170 COLOR0:IFF=0THEN1220
1180 SW=SW+1:ON SW GOTO1160,1190,1160,1200,1160,1210
1190 F1=F1+S1:X=P(P1+1):LOCATEP(P1),3:PRINT"      ":ON F1 GOTO810,840,870,900,930
,960,960,990,990
1200 F2=F2+S2:X=P(P2+1):LOCATEP(P2),5:PRINT"      ":ON F2 GOTO820,850,880,910,940
,970,970,1020,1020
1210 SW=0:F3=F3+S3:X=P(P3+1):LOCATEP(P3),7:PRINT"      ":ON F3 GOTO830,860,890,92
0,950,980,980,1050,1050
```

```
1220 SW=SW+1:ON SW GOTO1160,1230,1160,1250,1160,1270
1230 IFF1>1THENF1=F1-1:X=P(P1+1):LOCATEP(P1),3:PRINT"      ":ON F1 GOTO810,840,87
0,900,930,960,960,960
1240 RETURN
1250 IFF2>1THENF2=F2-1:X=P(P2+1):LOCATEP(P2),5:PRINT"      ":ON F2 GOTO820,850,88
0,910,940,970,970,970
1260 RETURN
1270 SW=0:IFF3>1THENF3=F3-1:X=P(P3+1):LOCATEP(P3),7:PRINT"      ":ON F3 GOTO830,8
60,890,920,950,980,980,980
1280 RETURN
1290 LINE(FL,8)-(FL+20,9),PSET,5,BF:L=L-10:FL=FL+20:IFFL=460ORFL=478THENC=6:GET@
(300,171)-(300,171),H%,G,2:IFH%(0)<0THENPAINT(300,171),6,0:RETURN:ELSERETURN
1300 IFFL=580ORFL=578THENC=7:GET@(300,171)-(300,171),H%,G,6:IFH%(0)<0THENPAINT(3
00,171),7,0:RETURN:ELSERETURN
1310 IFFL>637THEN390:ELSEIFFL=160THENS3=2:ELSEIFFL=320THENS2=2:ELSEIFFL>477THENS
1=2:ELSERETURN
1320 RETURN
1330 LINE(0,177)-(639,199),PSET,6,BF
1340 CONNECT(290,168)-(288,169)-(288,170)-(246,168)-(286,171)-(288,171)-(290,172
)-(310,172)-(312,171)-(314,171)-(354,168)-(312,170)-(312,169)-(310,168),0:CONNEC
T(280,170)-(280,175)-(282,176),:CONNECT(320,170)-(320,175)-(318,176)
1350 CONNECT(292,159)-(308,159)
1360 LINE(290,168)-(310,168),PSET:CONNECT(292,167)-(292,164)-(298,161)-(302,161)
-(308,164)-(308,167),1:PAINT(300,165),1,C,1:PAINT(300,171),C,0:LINE(300,159)-(30
0,168),PSET,0:CONNECT(298,167)-(298,165)-(302,165)-(302,167),0
1370 LOCATE28,13:PRINT"Player ";:PRINTPL;:LOCATE28,15:PRINT"Barrier";:PRINTB1-1;
:LOCATE1,13:PRINT"Score";:LOCATE1,15:PRINT"HighScore";:PRINTHS;:SYMBOL(95,125),P
0$,2,2,1
1380 X=614:FORI=1TO9:LINE(X,169)-(X+24,176),PSET,5,BF:X=X-30:GOSUB1440:NEXT
1390 LOCATE30,0:PRINT"FUEL";:Y=639:FORI=1TO60:LINE(Y,8)-(639,9),PSET,7,BF:Y=Y-1:
NEXT:FORI=1TO60:LINE(Y,8)-(580,9),PSET,6,BF:Y=Y-2:NEXT:FORI=1TOL:LINE(Y,8)-(460,
9),PSET,2,BF:Y=Y-2:NEXT:IFL<=0THENFL=458
1400 FORI=1TO10:LINE(X,169)-(X+24,176),PSET,5,BF:X=X+30:GOSUB1440:NEXT:BEEP1:FOR
I=0TO50:NEXT:BEEP0:SYMBOL(240,70),"Hit 3 Key",2,1,1
1410 PAINT(300,171),1,0:PAINT(300,171),6,0:PAINT(300,171),3,0:A$=INKEY$:IFA$="3"
THENLINE(240,70)-(390,80),PSET,5,BF:BEEP1:FORI=0TO50:NEXT:BEEP0:SYMBOL(150,40),"
Fighter",6,3,2:SYMBOL(260,80),"Start",2,1,1:ELSE1410
1420 FORI=0TO50:PAINT(300,171),6,0:PAINT(300,171),1,0:PAINT(300,171),3,0:NEXT:LI
NE(150,40)-(486,90),PSET,5,BF:BEEP1:FORI=0TO50:NEXT:BEEP0:FORI=0TO2000:NEXT:X=17
7:FORI=1TO25:LINE(0,X)-(639,X),PSET,5:X=X+1:FORY=0TO15:NEXTY:NEXTI:FORI=0TO3000:
NEXT
1430 INTERVALON:KEY(2)ON:KEY(3)ON:KEY(4)ON:KEY(5)ON:C=2:GOTO240
1440 CONNECT(X+24,169)-(X+24,175)-(X+22,175)-(X+22,174)-(X+20,173)-(X+18,173)-(X
+16,174)-(X+16,175)-(X+10,175)-(X+10,174)-(X+8,173)-(X+6,173)-(X+4,174)-(X+4,175
)-(X,175)-(X,172)-(X+3,171)-(X+12,171)-(X+14,170)-(X+14,169)-(X+24,169),0
1450 CONNECT(X+6,176)-(X+8,176),:CONNECT(X+18,176)-(X+20,176),:PAINT(X+12,173),3
,0:RETURN
1460 COLOR0,5:CLS:B=0:B1=4:C=5:F1=0:F2=0:F3=0:G=296:Z=10:P1=1:P2=21:P3=41:PL=3:P
0=0:P0$="0":L=230:FL=0:S1=1:S2=1:S3=1:SW=0:T=0:ONINTERVALGOSUB1290:INTERVAL8:ONK
EY(2)GOSUB460:ONKEY(3)GOSUB430:ONKEY(4)GOSUB420:ONKEY(5)GOSUB440:GOSUB1330:GOTO3
30
```

FM-8

ギャラクシアン YAMAUCHIの秘密

59 41 4D 41 55 43 48 49

■COMPAC O.H

先月号のインベーダー・ゲームはいかがでしたか．FM-8でも高速ゲームができるという証明になったのではないでしょうか．

今回は，ゲーム・センターでおなじみの『ギャラクシアン』を紹介するとともに，FM-8の表示を高速化するためのノウハウも紹介します．

遊び方

ゲーム自体は知らない人はいないと思うので特に説明はしません．キーの操作は前回と同様に，1，2，3で，それぞれ，左，ストップ，右に砲台が動き，STOPキーでビームの発射，スペースキーで再ゲームです．

FM-8の高速化 —— YAMAUCHIの秘密

FM-8でゲームを作ろうとすると，必らず突き当る壁が『**V-RAMのアクセス**』です．FM-8は，PC，MZなどの他のマイコンと異なり，V-RAMがメインCPUの管理下になく，すべてサブCPUがその管理をしているので，画面に何か表示させるときはBIOSを使うか，またはサブCPUに直接コマンドを送ってやらなければなりません．

しかし，この方法ではメインCPUはマシン語により，スピード・アップするのに対して，サブCPUの表示速度はBASIC使用時と何ら変りなく，メインCPUはその大半の時間をサブCPUのWAITループに費やすことになります．

前回および今回のゲームは，サブCPUを少しでも速くするために，サブシステム内のROMを一切使わずに，メインCPUからサブCPUを動かすためのプログラムをあらかじめ転送し，サブCPUの管理をそのプログラムに移す方法を使っています．

サブCPUに対するコマンドはいくつか知られていますが，この中に『**メインテナンス・コマンド**』と言うのがあり，これはさらにサブコマンドを持っています．サブコマンドはサブコマンド・コードと複数個のオペランド（ない場合もある）で構成されています．また未定義のサブコードの場合は，このメインテナンス・コマンドを終了します．

これらのコマンドを使ってメインCPUからサブCPUへプログラムを転送し制御を移します．ただ，このとき注意しなければならないことは，プログラムの**転送先**と『**V-RAMアクセス・フラグ**』です．**転送先**はサブシステムのRAM上であれば，ほとんどどこでも構わないようですが，V-RAM上への転送は後述する『**V-RAMアクセス・フラグ**』をSETする必要があり，また＄D000以降はI/Oや割り込

写真1

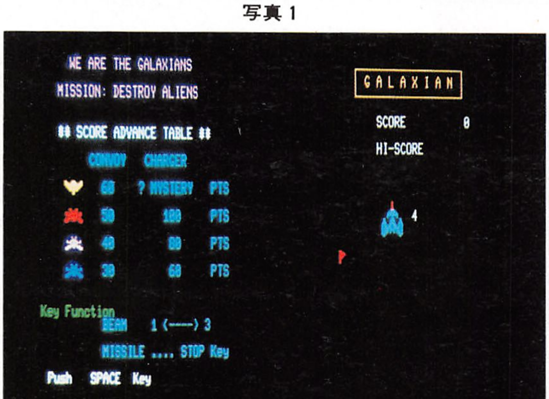


写真2



みに対するワーク・エリアなどがあるため，転送は避けた方が良いようです．

『V-RAMアクセス・フラグ』と言うのは文字通り，V-RAMをアクセスするときにSETするフラグで，＄D409番地をREADするとSET…つまりアクセス可能になり，WRITEするとRESET…アクセス不可能になります．通常，私達が使用するときは，一番最初にSETしておけば特に問題はないようです．

以上説明したコマンドを使って，サブシステムにプログラムを転送します．実際に私が行っている方法は最初に，『V-RAMアクセス・フラグ』をSETし，V-RAM上にプログラム転送のためのプログラムを転送してこのプログラムに制御を移し，ゲーム実行のためのプログラムを転送しています．なぜこのように面倒なことをやるのかと言うと，『メインテナンス・コマンド』のMOVEサブコマンドでは一度に120バイトたらずのプログラムしか転送できず，効率が悪い上にアドレスの計算が大変だからです．腕に自信のある方はMOVEサブコマンドを何回も使って結構です．

高速化のテクニック

サブCPUのメイン・プログラム(？)は，メインCPUとの共通領域の特定アドレスを常時読み込んで，それがコマンドであるかどうかを判断し，コマンドであれば，その処理ルーチンへジャンプします．たとえば，〝＄EE〞であれば〝ビーム砲の爆発パターンを表示する〞などです．

この点，サブCPUのモニタは汎用処理となっているのでアドレス，色の指定，表示すべきパターンなどを共通領域から読み取り計算しますが，この方法では色，表示すべきパターンはすでに自分の領域に持っているため，かなりのスピード・アップとなります．

おわりに

FM-8の表示速度はお世辞にも速いとは言えませんが，640×200ドットの分解能および1ドット単位での色指定は他のどんなマイコンにも負けないと思います．２つの6809を上手に使い，あなたもFMの高速プログラムに挑戦してみてください！

参考文献

1）芸夢狂人：〝ギャラクシアン〞，I/O，’80年９月号

2）芸夢狂人：〝PCGギャラクシアン〞，I/O，’81年１月号



図１　メインテナンス・コマンド



図２　メインテナンス・サブコマンド

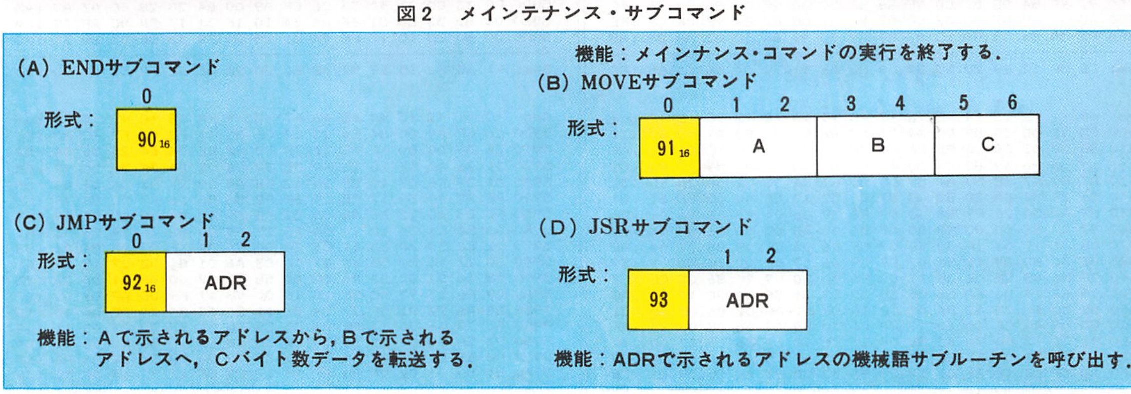


図３　サブシステムI/Oアドレス

アドレス	内容	機能
＄D401	キーデータ	READ＝キーデータ
＄D409	V-RAMアクセス・フラグ	READ＝SET WRITE＝RESET

ギャラクシアン・マシン語ダンプ・リスト

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
5000	10	CE	6F	FF	86	20	1F	8B	30	8D	01	7E	BF	01	E1	17	:90
5010	0F	3C	8E	00	00	9F	00	9F	80	9F	82	17	01	42	86	35	:CD
5020	B7	FC	80	7F	FD	05	17	03	84	17	00	38	17	05	F0	17	:C4
5030	0A	92	17	01	3E	17	01	7C	17	05	EC	17	0A	86	17	02	:4E
5040	25	17	05	DB	17	05	D5	17	05	DD	17	0A	77	17	0A	08	:C7
5050	17	0A	71	17	08	D3	17	05	CE	17	0A	68	17	07	62	17	:8E
5060	06	D8	20	C5	96	02	84	0F	81	0A	27	01	39	96	8E	27	:25
5070	01	39	17	00	EB	86	20	B7	FC	80	7F	FD	05	17	00	E0	:8D
5080	B6	FC	85	7F	FD	05	81	31	27	09	81	33	27	5E	9E	84	:F5

5090	16	00	3C	9E	04	8C	37	30	26	01	39	A6	1F	26	05	A6	:DD
50A0	88	4F	27	02	03	06	86	55	A7	1F	A7	88	4F	6F	03	6F	:09
50B0	88	53	96	86	26	0F	A6	88	B0	27	02	03	06	6F	88	B1	:E4
50C0	86	55	A7	88	B0	30	1F	9F	04	9E	84	30	1F	9F	84	9E	:DE
50D0	84	17	00	8C	BF	FC	86	86	FF	B7	FC	82	86	50	B7	FC	:AB
50E0	80	96	86	B7	FC	81	7F	FD	05	16	00	54	9E	04	8C	37	:20
50F0	60	26	01	39	9E	04	A6	04	26	05	A6	88	54	27	02	03	:E5
Sum:	E9	90	ED	DF	94	92	75	EF	6D	86	BF	46	DF	15	5F	A9	:C3

GALAXIAN

ギャラクシアン　マシン語ダンプ・リスト

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
5100	06	86	55	A7	04	A7	88	54	6F	84	6F	88	50	96	86	26	:8B
5110	0F	A6	88	B2	27	02	03	06	6F	88	B1	86	55	A7	88	B2	:85
5120	30	01	9F	04	9E	84	30	01	9F	84	17	00	33	BF	FC	86	:D5
5130	7F	FC	82	86	50	B7	FC	80	96	86	B7	FC	81	7F	FD	05	:D7
5140	9E	04	CE	55	55	EF	84	EF	02	EF	88	50	EF	88	52	96	:A4
5150	86	27	01	39	86	55	A7	88	B1	39	7F	FD	05	7F	FD	05	:DD
5160	B6	FD	05	2B	F5	86	80	B7	FD	05	B7	FD	05	B6	FC	80	:82
5170	26	E8	39	96	86	27	01	39	96	8E	27	01	39	0F	0F	1C	:83
5180	BF	1A	40	96	0F	26	02	0F	8F	39	03	0F	96	8F	26	1D	:37
5190	9E	04	A6	89	FF	01	26	15	30	88	B1	9F	0C	86	DD	A7	:2A
51A0	84	9E	84	30	89	FD	81	9F	8C	03	86	03	8F	86	40	AA	:93
51B0	E4	A7	E4	3B	96	02	84	0F	81	08	27	01	39	96	86	26	:01
51C0	01	39	9E	0C	6F	84	8C	30	50	25	57	A6	88	B0	81	B0	:6E
51D0	27	4D	84	E0	81	E0	10	27	00	78	10	8E	20	12	34	10	:FC
51E0	30	88	B0	9F	FC	EE	A4	C6	03	11	93	FC	10	27	00	58	:8D
51F0	33	41	5A	26	F4	31	26	10	8C	20	3C	25	E8	35	10	86	:0F
Sum:	14	EB	85	6D	7C	7E	F6	41	04	6B	6A	5C	95	96	EF	CC	:3D

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
5200	DD	30	88	B0	A7	84	9F	0C	9E	8C	30	89	FD	80	9F	8C	:A6
5210	17	FF	4D	BF	FC	86	86	DD	B7	FC	80	7F	FD	05	39	0F	:03
5220	86	39	17	FF	3B	9E	8C	BF	FC	86	86	80	B7	FC	80	7F	:33
5230	FD	05	0F	86	9E	84	17	FF	27	BF	FC	86	7F	FC	81	86	:B9
5240	50	B7	FC	80	7F	FD	05	39	35	10	AE	A4	A6	23	97	4C	:80
5250	20	11	30	88	B0	A6	82	84	E0	81	E0	27	F8	30	01	A6	:7C
5260	84	97	CC	9F	10	39	96	02	84	7F	81	41	27	01	39	17	:A4
5270	00	FF	9E	88	DE	08	96	8A	26	04	30	01	20	02	30	1F	:F7
5280	17	FE	DD	9F	88	BF	FC	86	96	8A	B7	FC	82	D6	0B	F7	:87
5290	FC	81	8E	FC	91	C6	06	34	54	C6	09	A6	C4	84	E0	81	:0A
52A0	E0	26	04	A7	80	20	02	6F	80	33	44	5A	26	ED	35	54	:AF
52B0	30	88	10	33	C8	50	5A	26	DE	86	E0	B7	FC	80	7F	FD	:86
52C0	05	03	0B	9E	08	96	8A	10	26	00	54	C6	06	34	04	C6	:2D
52D0	09	A6	84	84	E0	81	E0	26	18	A6	03	81	DD	26	0C	34	:A3
52E0	10	A6	84	97	CC	30	01	9F	10	35	10	A6	84	A7	03	6F	:05
52F0	84	30	04	5A	26	DB	35	04	30	88	2C	5A	26	CF	9E	08	:25
Sum:	30	77	27	AB	D4	27	79	18	FD	4D	E8	15	0A	6A	2A	02	:EC

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
5300	30	01	9F	08	C6	29	8E	40	00	EE	84	27	0C	6C	03	A6	:4F
5310	03	81	30	26	04	86	FF	97	8A	30	04	5A	26	EB	39	C6	:22
5320	06	34	04	C6	09	A6	84	84	E0	81	E0	26	18	A6	1F	81	:80
5330	DD	26	0C	34	10	A6	84	97	CC	30	1F	9F	10	35	10	A6	:C9
5340	84	A7	1F	6F	02	30	04	5A	26	DB	35	04	30	88	2C	5A	:C1
5350	26	CF	9E	08	30	1F	9F	08	C6	29	8E	40	00	EE	84	27	:E7
5360	0A	6A	03	A6	03	81	04	26	02	0F	8A	30	04	5A	26	ED	:07
5370	39	9E	90	26	01	39	17	FD	E7	BF	FC	86	86	16	B7	FC	:52
5380	80	7F	FD	05	8E	00	00	9F	90	39	30	89	D0	00	CE	80	:CE
5390	00	1F	10	8E	00	50	9F	FE	33	C9	02	80	93	FE	24	F8	:D5
53A0	33	C9	FD	80	D3	FE	DF	FE	D3	FE	1F	01	39	8D	04	17	:F9
53B0	00	A8	39	8E	20	08	6F	80	8C	20	80	26	F9	8E	20	84	:03
53C0	6F	80	8C	50	00	26	F9	0F	06	8E	37	42	9F	04	8E	B9	:F0
53D0	92	9F	84	8E	80	0A	9F	88	8E	30	0A	9F	08	86	E3	30	:FC
53E0	0C	A7	08	A7	80	A7	08	A7	80	A7	08	A7	84	9E	08	C6	:FE
53F0	05	30	88	58	86	E2	A7	80	A7	80	A7	84	30	02	5A	26	:A8
Sum:	C8	5F	12	E9	20	13	87	50	E8	A6	91	7C	04	5B	E1	E5	:EC

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
5400	F5	9E	08	30	89	00	A4	C6	07	86	E1	A7	80	A7	80	A7	:21
5410	84	30	02	5A	26	F5	C6	09	9E	08	30	89	00	F0	86	E0	:AF
5420	A7	88	50	A7	89	00	A0	A7	80	A7	88	50	A7	89	00	A0	:C5
5430	A7	80	A7	88	50	A7	89	00	A0	A7	84	30	02	5A	26	E0	:33
5440	86	29	97	0A	30	8D	08	87	C6	52	10	8E	40	00	EE	81	:01
5450	EF	A1	5A	26	F9	86	30	97	8E	39	86	04	97	03	86	01	:C8
5460	97	07	8E	00	00	9F	80	0F	83	30	8D	09	06	03	CE	00	:00
5470	0B	01	17	01	1A	30	8D	09	0F	86	03	CE	09	03	17	01	:8E
5480	0E	30	8D	09	1B	86	07	CE	09	06	17	01	02	30	8D	09	:39
5490	29	86	05	CE	0E	08	17	00	F6	30	8D	09	2E	86	05	CE	:F2
54A0	10	0A	17	00	EA	86	05	30	8D	09	36	CE	10	0C	17	00	:A3
54B0	DE	30	8D	09	42	86	05	CE	10	0E	17	00	D2	86	05	30	:01
54C0	8D	09	4A	CE	10	10	17	00	C6	30	8D	09	56	CE	06	13	:AE
54D0	86	04	17	00	BA	30	8D	09	57	CE	10	14	86	05	17	00	:0C
54E0	AE	30	8D	09	5E	CE	10	16	86	05	17	00	A2	30	8D	09	:D0
54F0	68	CE	07	18	86	07	17	00	96	30	8D	09	6E	CE	39	01	:CB
Sum:	2C	A3	C2	B9	CE	2D	CB	97	80	9D	75	17	0D	1F	4B	7C	:43

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
5500	86	06	17	00	8A	30	8D	09	76	CE	39	02	86	06	17	00	:15
5510	7E	30	8D	09	7E	CE	39	03	86	06	17	00	72	30	8D	09	:A7
5520	86	CE	3D	05	86	07	17	00	66	30	8D	09	8B	CE	3D	07	:03
5530	86	07	17	00	5A	8E	99	0A	17	00	85	73	FC	91	7F	FD	:47
5540	05	8E	9B	8A	17	00	79	73	FC	A1	7F	FD	05	8E	9E	0A	:0F
5550	17	00	6D	73	FC	B1	7F	FD	05	8E	A0	8A	17	00	61	73	:C8
5560	FC	C1	7F	FD	05	17	00	6E	17	00	A0	17	FB	F2	86	20	:24
5570	B7	FC	80	7F	FD	05	17	FB	E7	B6	FC	85	7F	FD	05	81	:E6
5580	20	26	E8	17	FB	DA	86	36	B7	FC	80	7F	FD	05	39	34	:F7
5590	02	17	FB	CC	35	02	B7	FC	81	86	34	B7	FC	80	10	8E	:D6
55A0	FC	91	A6	80	A7	A0	26	FA	1F	30	86	08	3D	86	50	3D	:47
55B0	C3	80	00	1F	01	1F	30	30	86	BF	FC	86	7F	FD	05	39	:63
55C0	17	FB	9D	BF	FC	86	8E	FC	90	6F	80	8C	FC	FF	26	F9	:9F
55D0	86	E0	B7	FC	80	39	17	FB	87	8E	9E	3E	BF	FC	86	86	:9C
55E0	50	B7	FC	80	7F	FC	81	7F	FD	05	17	FB	73	96	03	8B	:A9
55F0	30	B7	FC	91	7F	FC	92	8E	9E	43	BF	FC	86	86	34	B7	:A2
Sum:	DD	ED	D4	D5	4F	B2	D0	4F	07	9F	47	26	7E	31	6B	24	:E4

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
5600	FC	80	86	07	B7	FC	81	7F	FD	05	39	17	FB	52	96	07	:F8
5610	B7	FC	81	86	39	B7	FC	80	7F	FD	05	39	0C	02	39	8E	:B5
5620	00	64	30	1F	26	FC	39	DE	10	26	01	39	9E	90	27	03	:B4
5630	17	FD	3E	C6	04	8E	20	0C	30	06	11	A3	84	10	27	00	:7B
5640	92	5A	26	F4	C6	02	8E	20	24	30	06	11	A3	84	10	27	:45
5650	00	7F	5A	26	F4	8E	20	36	11	A3	84	27	05	DC	10	16	:3D
5660	00	7B	0F	4C	C6	14	86	32	0D	4D	27	0C	C6	28	86	34	:9D
5670	0A	4D	27	04	C6	50	86	38	9E	80	3A	9F	80	34	02	1F	:22
5680	10	17	05	9A	8E	8C	C7	BF	FC	86	7F	FD	05	17	FA	D0	:4A
5690	35	02	B7	FC	91	CE	30	30	FF	FC	92	7F	FC	94	86	34	:FF
56A0	B7	FC	80	86	07	B7	FC	81	9E	10	17	FC	DD	9F	CE	BF	:BE
56B0	FC	86	7F	FD	05	86	10	97	CD	CE	00	00	DF	10	DE	3A	:D2
56C0	0F	36	0F	37	34	40	8D	47	35	06	0F	4C	0A	0A	16	FB	:8E
56D0	63	0C	4D	EE	04	34	40	6F	84	6F	01	20	0F	DC	10	93	:33
56E0	08	34	06	9E	10	CE	00	00	EF	81	6F	84	8D	21	35	40	:44
56F0	CE	00	00	9E	10	DF	10	17	FC	90	17	FA	63	BF	FC	86	:C3
Sum:	A6	8F	48	56	E3	E9	70	7D	A6	B4	F9	71	DD	D0	48	79	:BE

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
5700	9F	90	86	F1	B7	FC	80	7F	FD	05	0A	0A	16	FB	25	8E	:32
5710	3F	FC	30	04	EE	84	11	A3	62	26	F7	CE	00	00	EF	84	:55
5720	0F	86	6F	9F	20	0C	17	FA	37	9E	8C	BF	FC	86	86	80	:88
5730	B7	FC	80	7F	FD	05	17	07	9F	39	96	02	84	7F	81	45	:0B
5740	27	01	39	CE	45	00	C6	19	A6	C0	27	07	33	42	5A	26	:DC
5750	F7	20	14	17	00	40	84	2F	1F	89	86	80	ED	C4	17	00	:AB
5760	35	84	03	27	F9	A7	5F	C6	19	CE	45	00	10	8E	FC	90	:FE
5770	17	F9	ED	AE	41	8C	BE	80	25	02	6F	C4	A6	C0	A7	A0	:BD
5780	AE	C4	30	89	02	80	AF	C1	AF	A1	5A	26	E6	86	A0	B7	:B0
5790	FC	80	7F	FD	05	39	96	82	D6	82	9B	82	97	82	9B	82	:F9
57A0	D7	82	9B	82	4C	97	82	39	86	01	97	0E	96	0A	81	1D	:7E
57B0	24	02	0C	0E	81	0D	24	02	0C	0E	81	06	24	02	0C	0E	:D5
57C0	39	96	02	84	7F	81	40	27	01	39	96	8E	27	01	39	8D	:08
57D0	D7	10	8E	20	12	8E	40	00	C6	22	17	00	7E	17	FF	B6	:BE
57E0	81	F0	24	01	39	9E	2A	26	FB	9E	30	26	F7	9E	36	26	:9D
57F0	F3	BE	40	9C	27	22	10	8E	20	36	8E	40	9C	86	01	97	:52
Sum:	37	C8	2C	24	06	30	CB	0A	31	7C	FC	94	DB	A4	66	91	:0D

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
5800	0E	C6	01	17	00	55	10	8E	20	2A	8E	40	88	86	02	97	:9E
5810	0E	C6	03	17	00	45	20	25	BE	40	A0	27	29	10	8E	20	:24
5820	36	8E	40	A0	86	01	97	0E	C6	01	17	00	2E	10	8E	20	:9A
5830	2A	8E	40	90	86	02	97	0E	C6	03	17	00	1E	0F	4D	9E	:AD
5840	AE	9F	B4	9F	BA	39	10	8E	20	30	8E	40	88	86	05	97	:F9
5850	0E	C6	02	17	00	05	9E	9F	BA	39	D7	8B	EC	84	27	C6	:CE
5860	07	30	04	0A	8B	26	F6	39	D3	08	1F	03	A6	C4	84	E0	:F0
5870	81	E0	26	ED	EC	A4	27	09	0A	0E	26	01	39	31	26	20	:23
5880	F3	A6	03	81	30	24	DA	A6	C4	A7	23	6F	C4	6F	41	6F	:D1
5890	42	34	50	1F	31	17	FA	F2	17	F8	C5	BF	FC	86	86	16	:CA
58A0	B7	FC	80	7F	FD	05	35	50	A6	02	4A	A7	22	A6	03	81	:1E
58B0	1B	24	2A	4A	A7	A9	00	82	33	C8	4F	EF	A4	34	10	1F	:C5
58C0	31	17	FA	C6	AF	A9	00	80	35	10	17	FE	C9	81	28	24	:D0
58D0	F9	33	8D	03	90	33	C6	EF	A9	00	84	20	28	4C	A7	A9	:45
58E0	00	82	33	C8	51	EF	A4	34	10	1F	31	17	FA	9C	AF	A9	:FA
58F0	00	80	35	10	17	FE	9F	81	28	24	F9	33	8D	03	6B	33	:A0
Sum:	F1	63	50	15	E9	57	3B	E1	D0	2A	AE	AE	ED	57	61	00	:10

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
5900	C6	EF	A9	00	84	E6	23	C4	0F	5C	17	FB	53	F7	FC	80	:EF
5910	EE	A9	00	80	FF	FC	86	7F	FD	05	EE	84	EF	24	30	04	:D2
5920	31	26	0A	0E	10	26	FF	35	39	96	02	84	3F	81	20	27	:35
5930	01	39	C6	07	10	8E	20	12	EE	A4	10	27	01	06	34	04	:DF
5940	AE	A9	00	80	17	F8	19	BF	FC	86	86	16	B7	FC	80	7F	:8E
5950	FD	05	6C	22	E6	A9	00	82	EE	A9	00	84	EB	C0	EF	A9	:FF
5960	00	84	C1	34	10	24	00	B3	E7	A9	00	82	4F	34	06	A6	:A1
5970	22	C6	50	3D	E3	E1	C3	30	00	ED	A4	1F	01	A6	84	81	:88
5980	DD	27	1E	81	55	10	27	00	C3	A6	01	81	DD	27	12	81	:B1
5990	55	10	27	00	B7	A6	02	81	55	10	27	00	AF	81	DD	26	:2B
59A0	09	A6	23	97	4C	9F	10	16	00	98	17	F9	DD	AF	A9	00	:57
59B0	80	A6	22	81	0F	27	04	81	11	26	5C	34	70	AE	A4	10	:1D
59C0	8C	20	36	26	13	EE	A4	33	C8	50	11	93	30	27	05	11	:09
59D0	93	2A	26	04	35	70	20	3F	30	88	51	10	8E	20	3C	C6	:B4
59E0	08	EE	A1	27	07	5A	26	F9	35	70	20	2B	31	3E	A6	A4	:E7
59F0	81	DD	26	02	0F	86	AF	A4	17	F9	8F	17	F7	62	BF	FC	:38
Sum:	16	87	A3	94	58	F6	7A	D5	71	15	ED	F5	33	24	5B	2C	:B7

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
5A00	86	30	89	FD	80	AF	A9	00	80	86	B0	A7	B4	B7	FC	80	:58
5A10	7F	FD	05	35	70	20	1A	81	19	26	16	CE	00	00	EF	A4	:97
5A20	EC	24	D3	08	1F	01	A6	23	A7	80	A7	80	A7	84	16	00	:63
5A30	11	E6	23	C4	0F	5C	17	F7	27	F7	FC	80	BF	FC	86	7F	:B1
5A40	FD	05	35	04	31	26	5A	10	26	FE	ED	39	03	06	34	70	:F3
5A50	17	00	71	35	70	16	FF	EA	96	02	84	1F	81	10	27	01	:20
5A60	39	10	8E	20	3C	AE	A1	26	07	10	8C	20	4C	25	F6	39	:0B
5A70	6F	84	30	88	50	8C	37	D0	24	2E	A6	84	81	DD	26	02	:90
5A80	0F	86	81	55	26	02	03	06	86	B0	A7	84	AF	3E	AE	A8	:40
5A90	7E	30	89	02	80	17	F6	C8	AF	A8	7E	86	C0	BF	FC	86	:EA
5AA0	B7	FC	80	7F	FD	05	20	C1	17	F6	B5	AE	A8	7E	30	89	:E4
5AB0	02	80	BF	FC	86	86	80	B7	FC	80	7F	FD	05	CE	00	00	:4B
5AC0	EF	3E	20	A5	D6	4C	26	04	D6	CC	27	3F	D6	4C	27	0C	:9B
5AD0	C4	03	CB	03	58	4F	97	4C	D3	80	DD	80	D6	CC	27	0B	:A3
5AE0	C4	03	CB	03	4F	97	CC	D3	80	DD	80	DC	80	17	01	2E	:99
5AF0	8E	8C	C7	BF	FC	86	7F	FD	05	96	83	26	0E	9E	80	8C	:9A
Sum:	09	D2	AE	1B	ED	FE	52	F1	C4	EE	6C	E7	C1	65	A7	D7	:7B

ギャラクシアン　マシン語ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5B00 01 F4 25 07 0C 03 03 83 17 FA CB 96 06 26 3A 96 :24
5B10 8E 26 0B 96 CD 26 13 96 0A 10 27 00 ED 39 96 02 :F0
5B20 84 7F 81 40 27 01 39 0A 8E 39 96 02 27 05 84 7F :BD
5B30 81 40 39 0A CD 27 01 39 17 F6 25 9E CE BF FC 86 :11
5B40 86 16 B7 FC 80 7F FD 05 39 9E 04 30 1F C6 06 6F :B5
5B50 88 50 6F 88 B0 6F 80 5A 26 F5 17 F6 03 9E 84 BF :D4
5B60 FC 86 86 EE B7 FC 80 7F FD 05 17 03 86 0F 06 C6 :25
5B70 03 9E 84 30 89 FD 80 17 F5 E6 BF FC 86 86 16 B7 :E1
5B80 FC 80 7F FD 05 30 03 17 F5 D6 BF FC 86 86 80 B7 :10
5B90 FC 80 7F FD 05 30 89 02 7D 5A 26 DB 86 30 97 8E :6B
5BA0 0A 03 27 2B 17 FA 2F C6 08 8E 37 42 9F 04 8E B9 :5E
5BB0 92 9F 84 96 86 26 01 39 6F 9F 20 0C 9E 8C 17 F5 :A1
5BC0 9F BF FC 86 86 80 B7 FC 80 7F FD 05 0F 86 39 86 :EE
5BD0 04 CE 0F 0B 39 8D 02 E9 17 F9 B4 9E 80 9C 00 25 :37
5BE0 10 9F 00 1F 10 17 00 36 8E 91 C7 BF FC 86 7F FD :CE
5BF0 05 8E 13 88 4A 26 FD 30 1F 26 F9 17 F5 62 86 36 :33
--------------------------------------------------------
Sum: ED BF E1 7C F4 02 3F B4 44 43 4B F9 DF 6C F0 19 :11
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6000 0A 35 83 00 10 8E FC 80 30 8D 00 0A C6 41 A6 80 :D0
6010 A7 A0 5A 26 F9 39 00 00 3F 59 41 4D 41 55 43 48 :40
6020 49 91 D3 96 3E 81 00 2C 93 3E 81 90 CE C0 00 7F :1D
6030 D3 80 7F D3 80 7F D3 80 7F D3 80 B6 D4 0A B6 D3 :E6
6040 80 27 F8 81 01 27 0D B7 D4 0A B7 D4 0A B6 D3 81 :89
6050 A7 C0 20 DB 7E C0 00 00 7F D3 80 10 CE CF FF 7F :9D
6060 D3 80 7F D3 80 B6 D4 0A B6 D3 80 27 F8 B7 D4 0A :76
6070 B7 D4 0A 30 8D 00 0C 30 02 A1 80 26 FA EC 84 30 :71
6080 8D 00 02 6E 8B E0 00 C1 50 01 E2 20 00 3C 80 00 :38
6090 45 F1 00 84 DD 02 70 16 00 5E 01 02 8A 02 02 B2 :C0
60A0 03 02 F8 04 03 20 A0 03 66 B0 03 C0 C0 03 DA EE :2B
60B0 03 E2 33 04 7A 34 04 AE 35 05 18 36 05 2D 39 05 :74
60C0 4A B6 D4 01 B7 D3 85 16 FF 8E BE D3 86 C6 08 4F :BB
60D0 A7 84 A7 89 C0 00 A7 89 80 00 30 88 50 5A 26 F0 :43
60E0 16 FF 75 BE D3 86 CE 00 00 4F C6 08 EF 89 80 00 :84
60F0 EF 89 C0 00 EF 81 A7 84 A7 89 80 00 A7 89 C0 00 :73
--------------------------------------------------------
Sum: 4C B8 AD 30 71 74 71 C8 9D C2 AB 49 2E 28 CC 38 :AC
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5C00 B7 FC 80 7F FD 05 17 F7 A4 39 17 F5 53 86 36 B7 :71
5C10 FC 80 7F FD 05 17 F7 9B 0C 07 17 F9 EE 39 34 06 :2A
5C20 17 F5 3D 8E 30 30 CE FC 91 AF C4 AF 42 AF 44 8E :77
5C30 30 00 AF 45 35 06 8E 27 10 8D 1A 8E 03 E8 8D 15 :E6
5C40 8E 00 64 8D 10 8E 00 0A 8D 0B C3 00 30 E7 C4 86 :E3
5C50 33 B7 FC 80 39 BF FC FE 6C C4 B3 FC FE 24 F9 6A :BC
5C60 C0 F3 FC FE 39 FE FF FF 00 00 02 01 01 00 00 00 :E6
5C70 01 01 02 02 02 01 01 01 00 00 FF 00 FF FF FE FE :04
5C80 FE FF FF FF 00 00 01 00 01 02 02 02 03 02 01 00 :09
5C90 00 FF FE 00 FF FE FE FE FD FE FF 00 00 01 02 00 :F3
5CA0 01 03 04 04 04 03 02 01 00 FF FD 00 FF FD FC FC :06
5CB0 FC FD FE FF 00 01 03 00 00 01 02 03 01 00 FF FE :FE
5CC0 FD FF 00 00 00 FF FE FD FF 00 01 02 03 01 00 01 :FD
5CD0 10 03 2A 00 F0 03 0A 00 BC 02 26 00 A4 02 0E 01 :D3
5CE0 60 04 2A 01 40 04 0A 01 B0 05 2A 01 90 05 0A 01 :5E
5CF0 0C 03 26 00 F4 03 0E 01 5C 04 26 01 44 04 0E 01 :19
--------------------------------------------------------
Sum: F0 23 C2 5F 12 A9 8A BB 0F 56 FA 31 32 6C 1A 4C :C8
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6100 30 88 4E 5A 26 E6 16 FF 4F BE D3 86 31 8D 07 24 :D0
6110 C6 08 EE A1 EF 89 C0 00 CE 00 00 EF 81 A6 A0 A7 :C0
6120 89 C0 00 4F A7 84 30 88 4E 5A 26 E6 C6 08 BE D3 :8E
6130 86 30 89 80 00 EE A1 EF 81 A6 A0 A7 84 30 88 4E :35
6140 5A 26 F2 16 FF 12 BE D3 86 CE D3 91 B6 D3 82 10 :FD
6150 26 00 8A 31 8D 05 9A 8D 51 30 89 C0 00 8D 4B BE :FA
6160 D3 86 30 89 C2 80 31 8D 05 CF CE D3 A1 8D 3B 31 :21
6170 8D 06 0E BE D3 86 30 89 C5 00 CE D3 B1 8D 2B 30 :70
6180 89 C0 00 8D 25 31 8D 06 40 CE D3 C1 BE D3 86 30 :A8
6190 89 87 80 8D 15 CE D3 D1 30 89 02 80 8D 0C CE D3 :19
61A0 E1 30 89 02 80 8D 03 16 FE AE 34 70 C6 09 A6 C0 :47
61B0 27 1E 34 74 C6 08 B6 D3 81 26 03 31 A8 18 6F 1F :6D
61C0 A6 A0 A7 80 EE A1 EF 84 30 88 4F 5A 26 F0 35 74 :8F
61D0 30 04 73 D3 81 5A 26 D6 73 D3 81 35 F0 31 8D 05 :00
61E0 10 8D 51 30 89 C0 00 8D 4B BE D3 86 30 89 C2 80 :51
61F0 31 8D 05 45 CE D3 A1 8D 3B 31 8D 05 84 BE D3 86 :70
--------------------------------------------------------
Sum: 16 85 2C B0 23 20 2F 20 A5 00 CD F5 87 4D E0 7C :A0
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5D00 AC 05 26 01 94 05 0E 00 B8 02 22 00 A8 02 12 01 :1B
5D10 08 03 22 00 F8 03 12 01 58 04 22 01 48 04 12 01 :19
5D20 A8 05 22 01 98 05 12 00 B4 02 1E 00 AC 02 16 00 :17
5D30 B0 02 1A 01 04 03 1E 01 00 03 1A 00 FC 03 16 01 :26
5D40 4C 04 16 01 50 04 1A 01 54 04 1E 01 A4 05 1E 01 :15
5D50 A0 05 1A 01 9C 05 16 00 58 01 12 00 5C 01 16 00 :55
5D60 26 01 1A 00 64 01 1E 00 68 01 22 00 6C 00 16 00 :AB
5D70 14 00 1E 57 45 20 41 52 45 20 54 48 45 20 47 41 :6F
5D80 4C 41 58 49 41 4E 53 00 4D 49 53 53 49 4F 4E 3A :6C
5D90 20 44 45 53 54 52 4F 59 20 41 4C 49 45 4E 53 00 :26
5DA0 2A 2A 20 53 43 4F 52 45 20 41 44 56 41 4E 43 45 :02
5DB0 20 54 41 42 4C 45 20 2A 2A 00 43 4F 4E 56 4F 59 :DA
5DC0 20 20 20 43 48 41 52 47 45 52 00 36 30 20 20 20 :22
5DD0 20 3F 20 4D 59 53 54 45 52 59 20 20 20 50 54 53 :13
5DE0 00 35 30 20 20 20 20 20 20 20 20 31 30 30 20 20 :36
5DF0 20 20 20 50 54 53 00 34 30 20 20 20 20 20 20 20 :9B
--------------------------------------------------------
Sum: 82 D0 7A 8D F6 75 B9 FD BB E7 A8 32 A6 32 C8 D0 :66
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6200 30 89 C5 00 CE D3 B1 8D 2B 30 89 C0 00 8D 25 31 :E4
6210 8D 05 B6 CE D3 C1 BE D3 86 30 89 87 80 8D 15 CE :F1
6220 D3 D1 30 89 02 80 8D 0C CE D3 E1 30 89 02 80 8D :C2
6230 03 16 FE 24 34 70 C6 09 A6 C0 27 1E 34 74 C6 08 :CF
6240 B6 D3 81 26 03 31 A8 18 A6 A0 A7 80 EE A1 EF 81 :90
6250 6F 84 30 88 4D 5A 26 F0 35 74 30 04 73 D3 81 5A :66
6260 26 D6 73 D3 81 35 F0 BE D3 86 31 8D 04 33 C6 10 :CA
6270 B6 D3 82 26 40 4F A7 1F A7 89 7F FF EE A1 EF 89 :3B
6280 80 00 EF 81 EE A1 EF 89 80 00 EF 84 30 88 4E 5A :4A
6290 26 E3 B6 D3 81 10 26 FD BF BE D3 86 C6 08 31 8D :A8
62A0 04 3F 30 89 BD 81 6F 1F A6 A0 A7 84 30 88 50 5A :9B
62B0 26 F4 16 FD A3 EE A1 EF 89 80 00 EF 81 EE A1 EF :45
62C0 89 80 00 EF 81 4F A7 89 80 00 A7 84 30 88 4C 5A :01
62D0 26 E3 B6 D3 81 10 26 FD 7F BE D3 86 C6 08 31 8D :68
62E0 03 FF 30 89 BD 81 A6 A0 A7 80 6F 84 30 88 4F 5A :BA
62F0 26 F4 16 FD 63 BE D3 86 C6 10 31 8D 03 E3 30 89 :DA
--------------------------------------------------------
Sum: 3C E1 36 44 D9 51 92 9A 54 42 24 3D 60 D9 11 02 :30
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5E00 20 2Q 38 30 20 20 20 20 20 50 54 53 00 33 30 20 :C2
5E10 20 20 20 20 20 20 20 20 36 30 20 20 20 20 20 50 :56
5E20 54 53 00 4B 65 79 20 46 75 6E 63 74 69 6F 6E 00 :36
5E30 42 45 41 4D 20 20 20 20 31 20 3C 2D 2D 2D 2D 3E :14
5E40 20 33 00 4D 49 53 53 49 4C 45 20 2E 2E 2E 2E 20 :61
5E50 53 54 4F 50 20 4B 65 79 00 50 75 73 68 20 20 20 :8F
5E60 53 50 41 43 45 20 20 4B 65 79 00 98 95 95 95 95 :C1
5E70 95 95 95 95 95 95 95 95 95 95 95 95 95 99 00 96 :C0
5E80 20 47 20 41 20 4C 20 41 20 58 20 49 20 41 20 4E :45
5E90 20 96 00 9A 95 95 95 95 95 95 95 95 95 95 95 95 :4C
5EA0 95 95 95 95 95 9B 00 53 43 4F 52 45 20 20 20 20 :80
5EB0 20 20 20 20 20 20 30 00 48 49 2D 53 43 4F 52 45 :2A
5EC0 00 2A 20 20 47 20 41 20 4D 20 45 20 20 4F 20 56 :E9
5ED0 20 45 20 52 20 20 2A 00 8E 00 50 C6 20 86 41 8D :59
5EE0 0B C6 20 86 40 8D 05 30 1F 26 F0 39 B7 FD 03 5A :F8
5EF0 26 FD 39 86 14 4A 27 55 34 02 8D 04 35 02 20 F5 :CF
--------------------------------------------------------
Sum: 77 0B 2C 6B 2D DF 69 16 B0 7E 83 7B BA 84 79 93 :17
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6300 C0 00 A6 A0 A7 84 30 88 50 5A 26 F6 16 FD 49 BE :C9
6310 D3 86 30 89 80 00 31 8D 04 DF C6 08 A6 A0 AA 84 :75
6320 A7 80 A6 A0 AA 84 A7 80 A6 A0 AA 84 A7 84 30 88 :19
6330 4E 5A 26 E8 16 FD 21 BE D3 86 30 89 C0 00 31 8D :38
6340 04 6F C6 08 A6 A4 AA 89 C0 00 A7 89 C0 00 A6 A0 :B4
6350 AA 84 A7 80 A6 A4 AA 89 C0 00 A7 89 C0 00 A6 A0 :C8
6360 AA 84 A7 80 A6 A4 AA 89 C0 00 A7 89 C0 00 A6 A0 :C8
6370 AA 84 A7 84 30 88 4E 5A 26 CA 16 FC DB BE D3 86 :AD
6380 30 89 C0 00 31 8D 03 E1 C6 08 A6 A0 AA 84 A7 80 :18
6390 A6 A0 AA 84 A7 80 A6 A0 AA 84 A7 84 30 88 4E 5A :9A
63A0 26 E8 16 FC B3 BE D3 86 30 89 C0 00 31 8D 03 71 :95
63B0 C6 08 A6 A4 AA 89 40 00 A7 89 40 00 A6 A0 AA 84 :6F
63C0 A7 80 A6 A4 AA 89 40 00 A7 89 40 00 A6 A0 AA 84 :C8
63D0 A7 80 A6 A4 AA 89 40 00 A7 89 40 00 A6 A0 AA 84 :C8
63E0 A7 84 30 88 4E 5A 26 CA 16 FC 6D C6 1A 8E D3 8E :C9
63F0 30 02 5A 10 27 FC 61 A6 80 27 F5 81 01 27 29 81 :B5
--------------------------------------------------------
Sum: 71 FA 59 41 07 35 38 BF 5E FC 00 0D 56 0D 0B A3 :B0
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5F00 33 8D 00 33 8E 00 0F 27 15 E6 C0 27 11 4F 1F 02 :1A
5F10 1F 10 86 40 8D 1A 30 1F 27 EA 8D 03 20 E6 39 1F :EA
5F20 20 86 41 8D 0B 1F 20 86 40 8D 05 30 1F 20 F0 39 :B4
5F30 B7 FD 03 5A 26 FD 39 80 70 60 50 40 30 20 10 20 :CD
5F40 30 40 50 60 50 40 30 20 50 40 30 20 00 39 B6 FD :CC
5F50 05 2B FB 1A 50 86 80 B7 FD 05 8D 74 7F FD 05 8E :64
5F60 00 00 30 1F 26 FC B6 FD 05 2B FB 86 80 B7 FD 05 :0E
5F70 17 00 91 7F FD 05 8E 00 00 30 1F 26 FC B6 FD 05 :E0
5F80 2B FB 33 8D 00 D2 30 8D 0A 22 BF 20 00 C6 FF 86 :CB
5F90 80 B7 FD 05 B7 FD 05 A6 C0 B7 FC 81 F7 FC 80 7F :7E
5FA0 FD 05 7F FD 05 B6 FD 05 2B FB 86 80 B7 FD 05 B7 :D7
5FB0 FD 05 B6 FC 80 26 E8 11 B3 20 00 26 D2 86 01 B7 :5C
5FC0 FC 80 7F FD 05 7F FD 05 8E 00 00 30 1F 26 FC 39 :B6
5FD0 10 8E FC 80 30 8D 00 0A C6 21 A6 80 A7 A0 5A 26 :B5
5FE0 F9 39 00 00 3F 59 41 4D 41 55 43 48 49 93 D3 8F :B7
5FF0 90 34 03 1A 50 B7 D4 0A B6 D4 08 B6 D4 09 B6 D4 :75
--------------------------------------------------------
Sum: AF C2 B9 94 0F C4 B8 CF 31 9B AB CF DE C5 71 44 :B6
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6400 02 27 12 EE 81 33 C9 80 00 86 18 AA C4 A7 C4 6F :0C
6410 C9 FD 80 20 DD EE 81 33 C9 C0 00 86 18 AA C4 A7 :21
6420 C4 6F C9 FD 80 16 FF CA EE 81 86 18 AA C4 A7 C4 :3E
6430 86 18 AA C9 C0 00 A7 C9 C0 00 6F C9 FD 80 6F C9 :EE
6440 BD 80 16 FF AD 31 8D 03 E3 C6 08 BE D3 86 30 89 :41
6450 C0 00 A6 A0 A7 84 30 88 50 5A 26 F6 16 FB F9 31 :EA
6460 8D 03 C1 C6 10 20 E4 CE 00 00 C6 18 BE D3 86 30 :1E
6470 89 FD 80 EF 81 EF 84 30 88 4E 5A 26 F6 C6 0F 34 :6E
6480 04 BE D3 86 31 8D 03 0C 30 89 BD 80 C6 18 34 10 :D0
6490 EE A1 EF 81 EE A1 EF 84 30 88 4E 5A 26 F2 35 10 :BE
64A0 C6 18 30 89 C0 00 EE A1 EF 81 EE A1 EF 84 30 88 :10
64B0 4E 5A 26 F2 8E 13 EC 30 1F 26 FC BE D3 86 30 89 :8E
64C0 7D 80 34 10 31 8D 03 9C C6 18 EE A1 EF 81 EE A1 :0A
64D0 EF 84 30 88 4E 5A 26 F2 35 10 30 89 40 00 C6 18 :07
64E0 EE A1 EF 81 EE A1 EF 84 30 88 4E 5A 26 F2 8E 13 :1A
64F0 EC 30 1F 26 FC 35 04 5A 10 26 FF 83 16 FB 59 BE :D0
--------------------------------------------------------
Sum: F4 D1 8C E9 59 F9 FD 6C DB C3 BB 43 39 31 C0 7C :37
```

ギャラクシアン　マシン語ダンプ・リスト

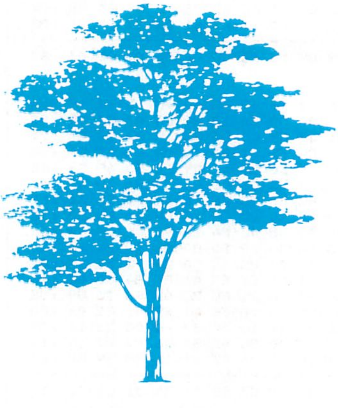
```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6500 D3 86 CE D3 91 8D 07 A6 C4 26 FA 16 FB 4A A6 C0 :6A
6510 C6 08 3D C3 D8 00 1F 02 C6 08 34 10 A6 A0 A7 84 :4A
6520 A7 89 C0 00 A7 89 80 00 30 88 50 5A 26 EE 35 10 :5B
6530 30 01 39 B6 D3 81 7F D3 82 84 01 27 03 73 D3 82 :BF
6540 B6 D3 81 7F D3 83 84 02 27 03 73 D3 83 B6 D3 81 :62
6550 7F D3 84 84 04 27 03 73 D3 84 BE D3 86 CE D3 91 :9B
6560 8D 07 A6 C4 26 FA 16 FA EF A6 C0 C6 08 3D C3 D8 :29
6570 00 1F 02 C6 08 34 10 A6 A0 7D D3 84 27 02 A7 84 :A1
6580 7D D3 83 27 04 A7 89 C0 00 7D D3 82 27 04 A7 89 :1B
6590 80 00 30 88 50 5A 26 DF 35 10 30 01 39 8E 00 00 :24
65A0 6F 89 40 00 6F 89 80 00 6F 80 8C 40 00 26 F1 16 :98
65B0 FA A6 8E 00 00 C6 36 6F 89 40 00 6F 89 80 00 6F :49
65C0 80 5A 26 F3 30 88 1A 8C 40 00 25 E9 16 FA 89 8E :C6
65D0 A5 B5 C6 1B 34 14 6F 89 80 00 6F 89 C0 00 6F 80 :A2
65E0 5A 26 F3 35 14 30 88 50 8C BE 80 25 E7 8E A5 B7 :84
65F0 7F D3 82 F6 D3 81 C1 0A 25 6E 31 8D 03 26 C6 20 :49
---------------------------------------------------------
Sum: 96 EE 93 C1 F6 0C 09 0D 63 5D 17 ED AB F4 60 37 :EA

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6600 34 10 30 89 C0 00 EE A1 EF 81 EE A1 EF 84 30 88 :76
6610 4E 5A 26 F2 AE E4 30 89 01 E0 10 8E D9 88 C6 08 :B9
6620 A6 A0 A7 84 A7 89 80 00 30 88 50 5A 26 F2 AE E4 :2D
6630 30 89 01 E1 10 8E D9 80 C6 08 A6 A0 A7 84 A7 89 :01
6640 80 00 30 88 50 5A 26 F2 35 10 30 05 7C D3 82 B6 :FB
6650 D3 82 81 05 26 07 7F D3 82 30 89 0C 67 F6 D3 81 :52
6660 C0 0A F7 D3 81 16 FF 8B B6 D3 82 27 10 C6 05 3D :FF
6670 FD D3 86 1F 10 B3 D3 86 C3 0C 80 1F 01 B6 D3 81 :0A
6680 10 27 F9 D4 31 8D 03 1C C6 08 34 10 A6 A0 A7 89 :69
6690 C0 00 30 88 50 5A 26 F4 35 10 30 02 7A D3 81 20 :A1
66A0 DC 00 1F 80 00 00 7F E0 00 01 FF F8 00 01 FF F8 :CA
66B0 00 01 06 08 00 30 0F 00 C0 30 1F 80 C0 78 1F 81 :B5
66C0 E0 DC 3F C3 B0 DC F9 F3 B0 DF F9 FF B0 DF FF FF :4A
66D0 B0 DF DF BF B0 DF 10 8F B0 FC 00 03 F0 30 00 00 :2A
66E0 C0 00 00 00 06 06 06 06 0F 00 00 00 00 00 00 00 :E7
66F0 00 60 3C 06 78 FF 1E 7E FF 7E 1F FF F8 07 99 E0 :C8
---------------------------------------------------------
Sum: 64 35 D4 CB 8B FC D2 76 3F B2 49 0B 01 C9 56 F3 :5F

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6700 01 FF 80 00 18 00 00 18 00 60 3C 06 78 FF 1E 7E :65
6710 FF 7E 1F FF F8 07 99 E0 01 FF 80 00 18 00 00 18 :C3
6720 00 00 18 00 00 18 00 00 7E 00 07 FF E0 1F 99 F8 :44
6730 7F FF FE 70 FF 0E 00 3C 00 01 81 80 00 C3 00 30 :2A
6740 FF 0C 3F 99 FC 01 FF 80 0F FF F0 7C 3C 3E 70 00 :C3
6750 0E 00 C3 00 60 FF 06 1F 99 F8 01 FF 00 3F FF FC :20
6760 7F 3C FE 30 00 0C 00 00 00 70 00 0E 7C 3C 3E 0F :78
6770 FF F0 01 FF 80 3F 99 FC 30 FF 0C 00 C3 00 01 81 :C3
6780 80 00 42 00 00 24 00 10 FF 08 1F 99 F8 00 FF 00 :AC
6790 0F FF F0 7F 18 FE 78 00 1E 00 24 00 18 FF 18 0F :8B
67A0 99 F0 00 FF 00 7F FF FE 3F 18 FC 0E 00 70 00 00 :D5
67B0 00 78 00 1E 7F 18 FE 0F FF F0 00 FF 00 1F 99 F8 :D8
67C0 10 FF 08 00 24 00 00 42 00 00 C3 00 00 66 00 30 :D6
67D0 FF 0C 3F 99 FC 00 FF 00 0F FF F0 7E 3C 7E 78 00 :8C
67E0 1E 00 66 00 30 FF 0C 1F 99 F8 60 FF 06 3F FF FC :0E
67F0 1F 3C F8 00 00 00 00 00 00 78 00 1E 7E 3C 7E 0F :30
---------------------------------------------------------
Sum: 7E 62 8D 6C D2 30 B7 4D 5A 45 93 4F BB 87 0A 8C :38

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6800 FF F0 00 FF 00 3F 99 FC 30 FF 0C 00 66 00 00 C3 :26
6810 00 10 38 10 CC 03 33 00 80 C8 30 D3 0C 0C 0C 10 :D9
6820 CA 30 21 07 00 00 00 00 00 00 00 00 00 00 3C 3C :9A
6830 3C 18 18 00 00 23 10 06 77 B0 07 FF FF 3F DE FF :ED
6840 FF DD FC 07 FF F0 0C 73 90 00 21 00 00 00 00 00 :FE
6850 00 00 00 23 00 03 FF F0 03 77 60 00 21 00 00 00 :10
6860 00 00 00 00 00 04 08 00 00 0E 31 00 00 1F 70 00 :DA
6870 01 BF F9 C0 00 FF FF 80 35 FF FF 00 1F FF FF 00 :47
6880 0F FF FF 80 07 FE FF 00 07 FC FF 80 0F F8 7F C0 :59
6890 07 F0 1F E0 0F F8 0F C0 1F F0 1F 00 7F F8 3F 80 :30
68A0 3F FA 7F C0 0F FE FF C0 1F FF FF 80 0F FF FE 00 :ED
68B0 07 FF FE 00 07 FB DE 00 0F F3 0C 00 09 A2 04 00 :A1
68C0 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :01
68D0 00 04 00 00 00 06 20 00 00 07 60 00 00 1F E4 00 :94
68E0 00 1F FC 00 00 FF FE 00 00 7F FE 00 00 FF FE 00 :92
68F0 00 7F FF 00 00 FF FE 00 00 7F FC 00 00 FF F8 00 :ED
---------------------------------------------------------
Sum: 62 6E FC 20 F7 4E F5 65 43 DE 77 D2 57 17 2F 4E :E0

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6900 01 FF F8 00 03 FF F8 00 03 FF E8 00 00 C7 80 00 :23
6910 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :02
6920 00 00 00 00 FF F0 00 00 FF FE 00 00 FF FF 80 00 :6A
6930 FF FF E0 00 FF FF F8 00 FF FF FE 00 FF FF FF 80 :4D
6940 FF FF FF E0 FF FF FF F8 FF FF FF FF FF FF FF FF :CA
6950 FF FF FF F8 FF FF FF C0 FF FF FF 00 FF FF F8 00 :A5
6960 FF FF E0 00 FF FF 00 00 FF FC 00 00 FF F0 00 00 :C6
6970 FF 80 00 00 FF 00 00 00 FF 00 00 00 FF 00 00 00 :7C
6980 FF 00 00 00 FF 00 00 00 FF 00 00 00 FF 00 00 00 :FC
6990 FF 00 00 00 FF 00 00 00 FF 00 00 00 FF 00 00 00 :FC
69A0 FF 00 00 00 E0 FC FF FC F0 C0 C0 C0 00 00 00 00 :06
69B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
---------------------------------------------------------
Sum: F9 7D B6 D8 DB E7 ED B4 EB B6 A4 BF F8 B3 F6 7F :8B
```



ALL CAST STAR TREK



FM-8

■粕谷昌朗

スター・トレックは，コンピュータ・シミュレーションゲームの最古参として，あまりにも有名ですが，いまも変らぬ人気があり，筆者のもっとも好きなゲームのひとつでもあります．

そこで，FM-8を手にしてから最初のプログラムとして，このゲームを作ることにしました．ゲーム内容そのものは，従来と，ほぼ同じですが，画面の方は，FM-8の多彩なグラフィック機能を，充分に生かしたつもりです．

ゲームの特徴

特に，画面関係は従来と異なっています．

❶カーク船長，Mr.スポック，Dr.マッコイなど，主要士官の顔が96×50ドットのグラフィックで表示される（ゲーム名の"ALL CAST"は，この特徴を示しています）．

❷USSエンタープライズの船内（いくぶん，宇宙戦艦ヤマトの影響を受けているようです）が表示され，その内でゲームが進行する．

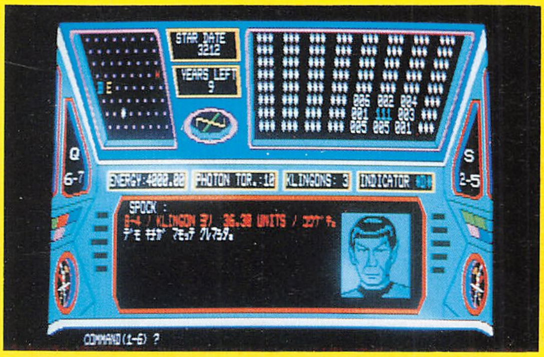
❸ショート・レンジ・センサー，銀河系マップの表示には SYMBOL文を使って立体感がでるように工夫してある．

❹SFの中によく出てくるが，筆者にはいまだに原理がわからない，『生命探知器』のコマンドがある．

遊び方

RUNさせると，まず，"FACEデータ"の読み込みを行なうので，テープレコーダーのプレイ・ボタンを押してください．

読み込みが終ると，ブザーが鳴り，タイトルが表示され，





エンタープライズ号と士官が勢揃いします．適当なキーをたたくと，ゲーム用の船内画面が描かれたあと，音を出すか否かを聞いてくるので，YまたはNで答えます．

次に，条件をプレーヤーが設定するか否かを聞いてきます．友人と同条件で競うときなどは，Yと答えると良いでしょう．自分で設定する場合は，クリンゴン数，年数，基地数を入力します．これらの入力値については，チェックをしていませんが，あまり極端な値を入れない方が良いでしょう．

ウーラ通信士が登場して，宇宙連邦からの入電を伝えます．条件が言いわたされ，OKかどうか聞いてくるので，YまたはNで答えます．Nの場合，新らたに条件が表示されます．なお，条件を自分で設定したときは，この問答はありません．

いよいよゲーム・スタートです．コマンドは，**表1**に示した6種です．ショート・レンジ・センサー，ロング・レンジ・センサー，銀河系マップ，ステータス・レポートは常に表示され，更新されるので，コマンドはありません．また，**図1**に画面の説明を示しました．

制限年数以内に，クリンゴンをすべてやっつければ勝ち，年数が過ぎたり，エネルギーがなくなると負けで，ゲーム・オーバーになります．勝ちのときは得点とクラスが表示されます．

もう一度やるか否かを聞いてくるので，YまたはNで答えます．

表1　コマンド表

コマンド	指定事項及び機能	コマンド取消法	備　考
1ワープ・エンジン（担当：スコット）	コース（COURSE 1～8，9）と速さ（WARP0，1～12，故障時は0，3まで）を指定．WARP 1で縦または横方向の1クオドラント（8セクタ）分の距離を移動．	コースまたは速さ指定時に，数字以外の文字を入力する（速さ指定時に0以下の数字を入力してもよい）．	●現クオドラント内の進向方向に障害物があれば緊急停止． ●連続使用は5回． ●エネルギー消費（速さに応じて）．（最低約 145UNITS 最高約1000UNITS）
2生命探知器（担当：スポック）	クリンゴンおよび基地に反応．ただし，たまに星にも反応したり，クリンゴンを見落したりする． 位置（セクタ）と保有エネルギーを表示する．	取消不可能	●基地および星の場合のエネルギーはクリンゴンと混乱するような値が表示される． ●エネルギー50UNITS消費．
3ダメージ・レポート（担当：船内コンピュータ）	船内機能のダメージ状況を表示する．ダメージの場合，修理完了年数も表示する．	取消不可能	●この機能が故障することはない．
4フェイザー　．（担当：チェコフ）	放射エネルギー量を指定． エネルギーは，各クリンゴンに等分配される．ただし，クリンゴンとの距離により減衰量が増す．	エネルギー指定時に，数字以外の文字または0以下の数字を入力する．	●連続使用は4回まで．
5光子魚雷（担当：ズールー）	コース（COURSE 1～8，9）を指定．命中した物を破壊するが，クリンゴンの場合，ダメージだけのときもある．また星の場合，新星発生もありうる．	コース指定時に数字以外の文字を入力する．	●連続使用は5回まで． ●クリンゴンに命中したが，ダメージのみのとき，そのクリンゴンの保有エネルギーは1/2になる．
6エネルギー変換器（担当：スコット）	光子魚雷1個当たり，300UNITSのエネルギーに変換する．変換する個数を指定する．	個数指定時に数字以外の文字または0以下の数字を入力する．	●エネルギーから光子魚雷への変換はできない．

注）ワープ・エンジン，フェイザー，光子魚雷の『連続使用』とは，そのコマンドだけを続けて使用すること．規定回数を過ぎると，『オーバーヒート』になるが，このとき，他のコマンド（3は除く）を1回でも使用すれば，オーバーヒート状態は解除になる．

プログラムについて

プログラムについては，スター・トレックとして，オーソドックスな構成なので説明を省略しますが，ジェネラル・フローチャート，変数表，行番号マップを載せたので参考にしてください（**表2**，**表3**，**図2**）．

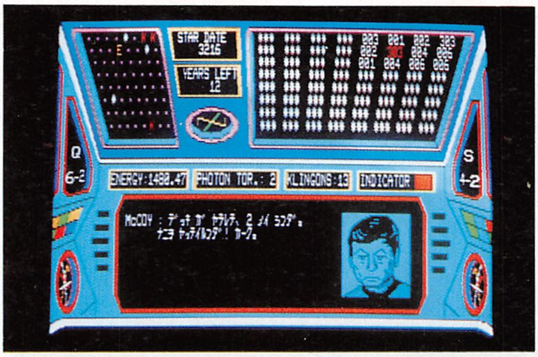
改造について

いまのままで，残りメモリが800バイト弱なので（文字領域は少ししか使っていないようなので，CLEAR文で文字領域を少なくすれば，少しメモリが増えます），あまり大きな改造はできないと思いますが，プログラムをもっと能率的にするなど，みなさんで工夫してみてください．

実を言うと，筆者は，初め士官の口や，まゆ毛を動かすアニメをやろうと思って，テスト・プログラムも作ったのですが，メモリを食う上，思ったより効果がなかったのでやめました（やはり，声が出ないと感じが悪いです）．

あとがき

マイコンでアニメを作るのは筆者の夢ですが，その，ほんの一歩として，本プログラムを作ったわけです．士官の顔については，トレッキーの皆さんの中には，いろいろと



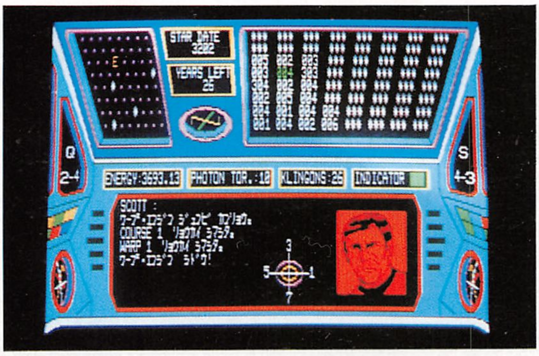
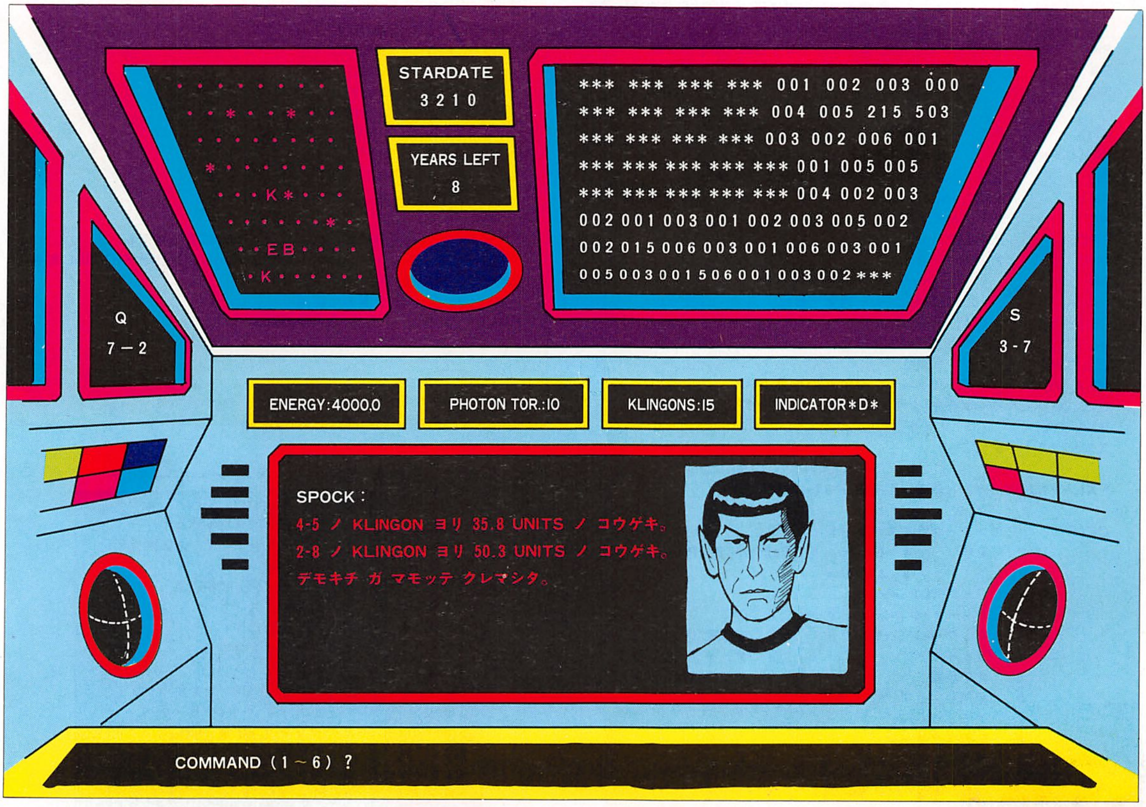


図1　画面説明図



画面説明図(付表)

❶ メイン・スクリーン
各士官が指令を要求する．また，各種情報が報告されたり，（特にスポックに）皮肉を言われたりする．

❷ コマンド・スクリーン
カーク船長(プレーヤー)の指令はすべて，ここで行なわれる．“ANSWER(Y/N)?”と“COMMAND(1～6)?”は1文字入力，他は適当な数字を入力してから[Return]

❸ ショート・レンジ・センサー・スクリーン
E：エンタープライズ，K：クリンゴン，B：基地，＊：星，・：何もなし(空間)従来と異なり，立体感を得るためゆがんで表示されているので，角度計算に注意．

❹ 銀河系マップ・スクリーン(ロング・レンジ・センサー兼用)
クオドラントの情報が3桁の数字で示される．未センスは＊＊＊，100の位：クリンゴン数，10の位：基地数，1の位：星の数，現在地はインジケーターと同色で，他は白で表示．

❺ ステータス・レポート表示ウィンドウ群
- ❺-1　宇宙歴(STARDATE)　(開始時は3200)
- ❺-2　残り年数
- ❺-3　クオドラント現在地
- ❺-4　セクター　〃
- ❺-5　残りエネルギー
- ❺-6　残り　光子魚雷数
- ❺-7　残り　クリンゴン数
- ❺-8　インジケーター
  残り年数，クリンゴン数，ダメージ状態などを総合的に評価して，Green，Yellow，Redの色表示をする．基地に帰港しているときは，水色で＊D＊の表示．

文句がある方がおられると思いますが，どうか，大目に見てください（縦方向のドット数が少ないので，目や口の表情をだすのが，難かしいのです）．

これからも，FM-8のグラフィック機能を生かしたプログラムを，いろいろと作っていこうと思います．

◘参考文献

1）米室九，船山大翼，〝特集STARTREK〞，ASCII，'78年1月号
2）〝TK-80BSでStar Trek〞，ASCII，'78年3月号
3）映画〝STAR TREK〞(パラマウント映画：松竹配給)パンフレット
4）ハヤカワ文庫「宇宙大作戦」シリーズ
5）TVシリーズ『スタートレック』(文献ではありませんが)

表2　変数および関数リスト(1)

指定のないものはすべて整数型

(I)　汎用変数

I, J, K, T	主にFOR文用
X, Y, X1, Y1, X2, Y2,	座標用
C1, C2	主にカウンタ用
CL	カラー指定用
A$	汎用文字変数

(II)　一　般

O	サウンド・フラグ(＝0：音なし，＝1音あり)
YR	制限年数
YEAR	残り年数
KL	初期クリンゴン数
KLG	残りクリンゴン数
MB	基地数，基地破壊数
ENG！	残りエネルギー
PT	残り魚雷数
SD	宇宙歴
DE	死者数
Q(*X*, *Y*)	クオドラント情報
QD(*X*, *Y*)	クオドラント情報(表示用)
QS(*x*, *y*)	クオドラント・センス・フラグ(＝0：未センス，=1：センス済)
SQ($)	QD(*x*, *y*)を文字型にしたもの
S$(*x*, *y*)	セクタ情報(E, K, B, ＊, ・)

QXE, QYE	エンタープライズのいるクオドラントの座標(0〜7)	
SXE, SYE	〃　セクタ　〃　(0〜7)	
	〃　クオドラントの情報(＝Q(QXE,QYE))	
SK	エンタープライズのいるクオドラント内の　リンゴン数	
SB	〃　基地数	
SS	〃　星数	
K(*i*)	クリンゴン位置(11〜88or0; *i*＝1〜5)	
KEI(*i*)	クリンゴン保有エネルギー(*i*＝1〜5)	
EI	攻撃エネルギー,汎用実数型変数としても使用	
AE！	攻撃エネルギー総量	
WC	ワープ連続使用回数	
PTC	光子魚雷　〃	
PHC	フェイザー　〃	
SC	ショート・レンジ・センサー・スイッチ (≒0：更新要, ＝1：更新不要)	
CS！	コース(1〜8.9)	
WL！	ワープ速さ(0.1〜12)	
DXI, DYI	1ステップ変位	
WSX, WSY	ワープ・スタート・ポイント(セクタ)	
QX2, QY2	ワープ行先(クオドラント)	
SX2, SY2	(セクタ)	
X3！, Y3！	変位	
FX1, FY1, FX2, FY2	顔表示用座標(PUT@文用)	

CE(*i*)　エンタープライズのコンディション

i＝0	全体	4：Green, 6：Yellow, 2：Red 5：Docked
i＝1	ワープ・エンジン	<0：OK ＝0：修理完了 >0：ダメージ(修理完了までの年数を示す)
2	ショート・レンジ・センサー	
3	ロング・レンジ・センサー	
4	フェイザー	
5	光子魚雷	
6	銀河系マップ	
7	メモリ・ルーム	
8	防御スクリーン	
9	生命探知器	
10	エネルギー変換器	

(III) 表示用文字変数

F$(*i*)　機能名

i＝0	"エンタープライズ"
1	"ワープ・エンジン"
2	"ショート・レンジ・センサー"
3	"ロング・レンジ・センサー"
4	"フェイザー"
5	"コウシ・ギョライ"
6	"ギンガケイ・マップ"
7	"メモリ・ルーム"
8	"ボウギョスクリーン"
9	"セイメイ・タンチキ"
10	"エネルギーヘンカンキ"

C$(*i*)　キャラクター名

i＝0	"KIRK："
1	"SPOCK："
2	"McCOY："
3	"SCOTT："
4	"SULU："
5	"CHEKOV："
6	"UHURA："
7	"ホシ"
8	"キチ"
9	"KLINGONセンカン"

(IV) 関　数

FNX(x, i)	画面作成のための，左右の対称位置を与える.
FNSX(x, y)	ショート・レンジ・センサ・スクリーンのX座標(SYMBOL文用)
FNMX(x, y)	銀河系マップ・スクリーン　〃　(　〃　)
FNSY(y)	上記2者のY座標　(　〃　)
FNL！($x_1 y_1 x_2 y_2$)	2点間の距離

表3 行番号マップ (II)メイン・ルーチン

10〜 60	宣　言　etc.
70〜 100	FACEデータ読み込み
110〜 140	タイトルおよびゲーム画面表示
150〜 300	条件設定
310〜 340	クオドラント初期設定
350〜 360	ステータス　〃
370〜 450	セクタ設定
460〜 470	ステータス判定
480	修理判定
490〜520	ロング・レンジ・センサー
530〜 560	ショート・レンジ・センサー
570〜 610	クリンゴン攻撃
620〜 720	故障発生，ステータス判定
730〜 850	コマンド入力
860〜1120	ワープ
1130〜1230	フェイザー
1240〜1340	光子魚雷
1350〜1430	生命探知器
1440〜1460	ダメージ・レポート
1470〜1510	エネルギー変換器
1520〜1590	ゲーム・オーバー処理

図2　ジェネラル・フローチャート(2)

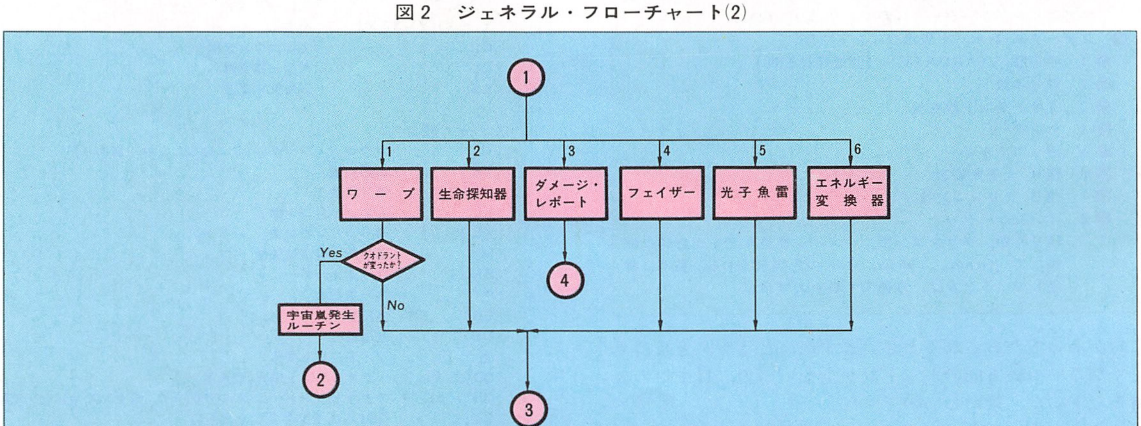
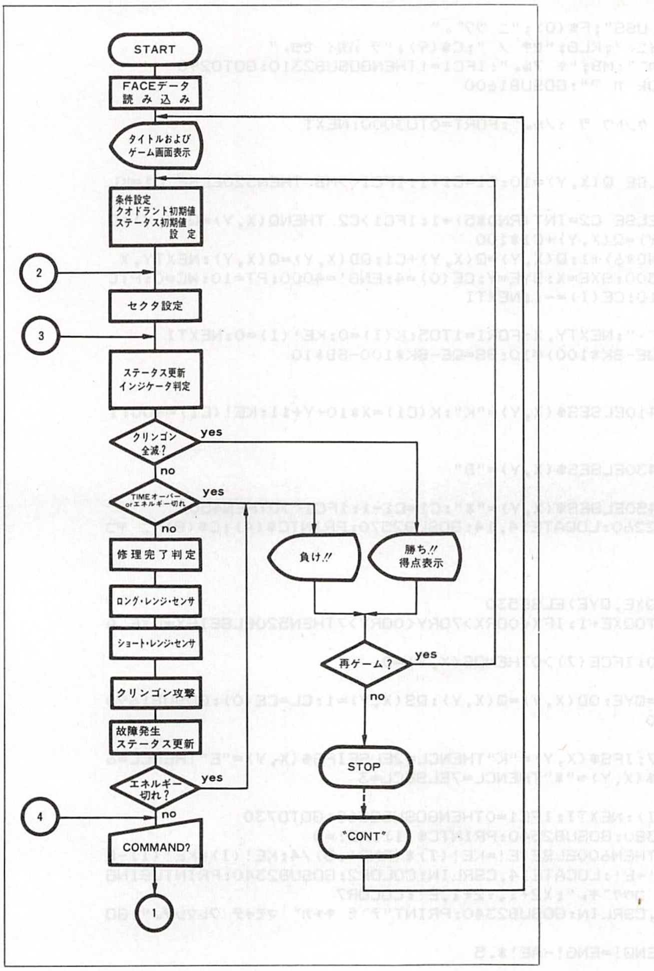


図2 ジェネラル・フローチャート(1)



(II)サブルーチン

1600〜1610	(Y/N)問答
1620〜1630	マップ表示
1640	マップ・クリア
1650〜1670	QD(x, y) →SQ$変換
1680	ショート・レンジ・センサ・クリア
1690	マップ 1クオドラント分 クリア
1700	〃 〃 表示
1710〜1740	基地帰港判定
1750〜1840	ステータス判定および表示
1850〜1860	故障数カウント
1870〜1930	故障発生
1940〜1950	死者〃
1960〜2060	修理完了判定
2070	ワープおよび魚雷の1STEP 変位計算
2080	〃 〃 イニシャライズ
2090	〃 〃 1STEP移動
2100〜2110	ショート・レンジ・センサ 1セクタ分,
2120	距離パラメータ計算
2130	クリンゴン破壊処理
2140〜2190	ワープおよび魚雷の移動表示
2200	連続使用回数初期化
2210〜2250	ダメージ・レポート表示
2260	メインスクリーン 全面 クリア
2270	〃 FACEエリア〃
2280	〃 メッセージ・エリア〃(1)
2290	〃 〃 〃(2)
2300	座標乱数発生
2310〜2330	タイマ 1〜3
2340〜2380	サウンド 1〜3
2390〜2410	メッセージ表示(オーバーロード)
2420	〃 (故障)
2430	〃 (準備完了)
2440〜2470	〃 (コース指定)
2480	〃 (緊急停止)
2490〜2500	〃 (命中)
2510	〃 (連邦より入電)
2520	〃 (成績評価)
2530〜2590	顔表示
2600〜2870	ゲーム画面表示
2880〜3020	タイトル表示

BASICプログラム・リスト

```
1 ' **  ALL CAST STAR TREK Ver.2.1  for FM-8   **
2 ' **  Programmed by M.Kasuya (Nov. 1981)      **
3 ' **  Copyright (C) 1981 by M.Kasuya          **
5 CLEAR,&H6F84:IF PEEK(&H6F85)<>&H8E OR PEEK(&H6F97)<>&H39 THEN LOADM"CASO:"
10 DEFINTA-Z:DEFFNSX(X,Y)=CINT((58-3*X)*(4*Y-196)/41+320):DEFFNSY(Y)=8*Y+8:DEFFN
MX(X,Y)=CINT((2-4*X)*(8*Y-392)/41+320):DEFFNL!(X1,Y1,X2,Y2)=1-.06*SQR((X1-X2)*(X
1-X2)+(Y1-Y2)*(Y1-Y2)):DEFFNX(X,I)=319+FIX(X*I+.5)
20 DIMFA(299)
30 DIM Q(7,7),QD(7,7),QS(7,7),S$(7,7),K(5),KE!(5),C$(9)
40 F$(0)="ｴﾝﾀｰﾌﾟﾗｲｽﾞ  ":F$(1)="ﾜｰﾌﾟ･ｴﾝｼﾞﾝ ":F$(2)="ｼｮｰﾄﾚﾝｼﾞ ｾﾝｻｰ  ":F$(3)="ﾛﾝｸﾞﾚﾝｼﾞ
ｾﾝｻｰ ":F$(4)="ﾌｪｲｻﾞｰ ":F$(5)="ｺｳｼ ｷﾞｮﾗｲ ":F$(6)="ｷﾞﾝｶﾞｹｲ ﾏｯﾌﾟ ":F$(7)="ﾒﾓﾘｰ ﾙｰﾑ
":F$(8)="ﾎﾞｳｷﾞｮ ｽｸﾘｰﾝ ":F$(9)="ｾｲﾒｲ ﾀﾝﾁｷ ":F$(10)="ｴﾈﾙｷﾞｰ ﾍﾝｶﾝｷ "
50 C$(0)="KIRK   : ":C$(1)="SPOCK : ":C$(2)="McCOY : ":C$(3)="SCOTT : ":C$(4)="SU
LU  : ":C$(5)="CHEKOV: ":C$(6)="UHURA : ":C$(7)="ﾎｼ ":C$(8)="ｷﾁ ":C$(9)="KLINGON
 ｾﾝｶﾝ "
60 WIDTH80,25:COLOR7,0:CLS
110 RANDOMIZETIME-INT(TIME/28800)*28800
120 O=0:CONSOLE24,1:GOSUB2880
130 LOCATE60,24:PRINT"PLEASE HIT ANY KEY";:FORT=0TO200:NEXTT:LOCATE60,24:PRINTSP
C(18);:IFINKEY$=""THEN130
140 GOSUB2600:SD=3200:FX1=424:FY1=119:FX2=519:FY2=168
150 GOSUB2260:LOCATE14,14:PRINT"ｻｳﾝﾄﾞ ﾂｷ ﾆ ｼﾏｽｶ ?":GOSUB1600:IFA$="N"THEN O=0ELS
E O=1
160 GOSUB2260:LOCATE14,14:PRINT"ｱﾅﾀ ｶﾞ ｼﾞｮｳｹﾝ ｦ ｷﾒﾏｽｶ ?":GOSUB1600:IFA$="N"THEN
C1=0:GOTO180ELSE C1=1
170 LOCATE7,24:GOSUB2340:INPUT"KLINGONS=";KLG:LOCATE7,24:GOSUB2340:INPUT"YEARS="
;YEAR:LOCATE7,24:GOSUB2340:INPUT"BASES=";MB
180 RANDOMIZETIME-INT(TIME/28800)*28800
190 FORX=0TO7:FORY=0TO7:QD(X,Y)=0:Q(X,Y)=0:QS(X,Y)=0:S$(X,Y)="･":NEXTY,X
200 GOSUB2510
210 GOSUB2260:IFC1=1THEN230
```

BASICプログラム・リスト

```
220 YEAR=RND*18+18:KLG=RND*31+10:IFABS(YEAR-KLG)>15THEN210ELSE MB=RND*2+1
230 YR=YEAR:KL=KLG
240 GOSUB2360:LOCATE14,15:PRINT" USS";F$(0);"ﾆ ﾂｸﾞ｡"
250 LOCATE14,16:PRINTYEAR;"ﾈﾝ ｲﾅｲﾆ､";KLG;"ｾｷ ﾉ ";C$(9);"ｦ ﾊｶｲ ｾﾖ｡"
260 LOCATE14,17:PRINT" ﾐｶﾀ ﾉ ｷﾁ ｶﾞ";MB;"ｷ ｱﾙ｡":IFC1=1THENGOSUB2310:GOTO290
270 LOCATE14,18:PRINT" ｼﾞｮｳｹﾝ ﾊ Ok ｶ ?":GOSUB1600
280 IFA$="N"THEN210
290 GOSUB2340:LOCATE14,19:PRINT" ｹﾝﾄｳ ｦ ｲﾉﾙ｡":FORT=0TO3000:NEXT
300 GOSUB2260
310 C1=0
320 GOSUB2300:IFQ(X,Y)>9THEN320ELSE Q(X,Y)=10:C1=C1+1:IFC1<>MB THEN320ELSE C1=KL
G
330 GOSUB2300:IFQ(X,Y)>99THEN330ELSE C2=INT(RND*5)+1:IFC1>C2 THENQ(X,Y)=Q(X,Y)+C
2*100:C1=C1-C2:GOTO330 ELSE Q(X,Y)=Q(X,Y)+C1*100
340 FORX=0TO7:FORY=0TO7:C1=INT(RND*6)+1:Q(X,Y)=Q(X,Y)+C1:QD(X,Y)=Q(X,Y):NEXTY,X
350 GOSUB2300:QXE=X:QYE=Y:GOSUB2300:SXE=X:SYE=Y:CE(0)=4:ENG!=4000:PT=10:WC=0:PTC
=0:PHC=0:SC=0:DE=0:MB=0:FORI=1TO10:CE(I)=-1:NEXTI
360 GOSUB1620
370 FORX=0TO7:FORY=0TO7:S$(X,Y)="･":NEXTY,X:FORI=1TO5:K(I)=0:KE!(I)=0:NEXTI
380 QE=Q(QXE,QYE):SK=QE¥100:SB=(QE-SK*100)¥10:SS=QE-SK*100-SB*10
390 S$(SXE,SYE)="E"
400 IFSK=0THENGOTO420ELSEC1=1
410 GOSUB2300:IFS$(X,Y)<>"･"THEN410ELSES$(X,Y)="K":K(C1)=X*10+Y+11:KE!(C1)=400:I
FC1<>SK THENC1=C1+1:GOTO410
420 IFSB=0THEN440
430 GOSUB2300:IFS$(X,Y)<>"･"THEN430ELSES$(X,Y)="B"
440 IFSS=0THEN460ELSEC1=SS
450 GOSUB2300:IFS$(X,Y)<>"･"THEN450ELSES$(X,Y)="*":C1=C1-1:IFC1<>0THEN450
460 GOSUB1750:IFCE(0)=5THENGOSUB2260:LOCATE14,14:GOSUB2570:PRINTC$(4);C$(8)"ﾆ ｷｺ
ｳ ｼﾏｼﾀ｡":GOSUB2310
470 GOSUB1770
480 GOSUB1960:GOSUB2320
490 IFCE(3)<=0THENQD(QXE,QYE)=Q(QXE,QYE)ELSE530
500 FORY=QYE-1TOQYE+1:FORX=QXE-1TOQXE+1:IFX<0ORX>7ORY<0ORY>7THEN520ELSEIFX=QXE A
NDY=QYE THENCL=CE(0):ELSECL=7
510 QS(X,Y)=1:GOSUB1690:GOSUB1700:IFCE(7)>0THENQS(X,Y)=0
520 NEXTX,Y
530 IFCE(2)>0THEN570ELSE X=QXE:Y=QYE:QD(X,Y)=Q(X,Y):QS(X,Y)=1:CL=CE(0):GOSUB1690
:GOSUB1700:IFCE(7)>0THENQS(X,Y)=0
540 IFSC=1THEN570
550 GOSUB1680:FORY=0TO7:FORX=0TO7:IFS$(X,Y)="K"THENCL=2ELSEIFS$(X,Y)="E"THENCL=6
ELSEIFS$(X,Y)="B"THENCL=5ELSEIFS$(X,Y)="*"THENCL=7ELSECL=3
560 A$=S$(X,Y):GOSUB2110:NEXTX,Y
570 SC=1:C1=0:FORI=1TO5:C1=C1+K(I):NEXTI:IFC1=0THENGOSUB2260:GOTO730
580 GOSUB2260:LOCATE15,14:GOSUB2380:GOSUB2540:PRINTC$(1):AE!=0
590 FORI=1TO5:GOSUB2120:IFK(I)=0THEN600ELSE E!=KE!(I)*(RND+.5)/4:KE!(I)=KE!(I)-E
!:E!=E!*FNL!(X1,Y1,X2,Y2):AE!=AE!+E!:LOCATE14,CSRLIN:COLOR2:GOSUB2340:PRINTUSING
"#-# ﾉ KLINGON ﾖﾘ ###.## UNITS ﾉ ｺｳｹﾞｷ｡";X2+1,Y2+1,E!:COLOR7
600 NEXTI:IFCE(0)=5THEN LOCATE14,CSRLIN:GOSUB2340:PRINT"ﾃﾞﾓ ｷﾁｶﾞ ﾏﾓｯﾃ ｸﾚﾏｼﾀ｡":GO
TO730ELSEGOSUB2320
610 ENG!=ENG!-AE!:IFCE(8)>0THEN ENG!=ENG!-AE!*.5
620 GOSUB1850:IFDM=10THEN720
630 IFAE!<60OR(AE!<100!ANDCE(8)<=0)THEN720
640 IFAE!>=150!ANDCE(8)>0THEN GOSUB1940:GOSUB2310:GOSUB1870:GOTO720
650 IFAE!>=100!ANDCE(8)>0THEN GOSUB1940:GOSUB2310:GOTO700
660 IFAE!<100THEN710
670 IFAE!<300ANDDM<6THEN710
680 IFAE!<400ANDDM<3THEN700
690 IFAE!>=400THEN700ELSE720
700 IFRND<.7THENGOSUB1870:GOTO720ELSE720
710 IFRND<.5THENGOSUB1870
720 GOSUB1770
730 GOTO760
740 GOSUB2310
750 GOSUB2260
760 C1=0
770 I=0
780 GOSUB2340
790 LOCATE7,24:PRINT"COMMAND(1-6) ?";:FORT=0TO220:NEXTT:LOCATE7,24:PRINTSPC(14);
:A$=INKEY$:IFA$<>""THEN810
800 I=I+1:IFI=50 AND C1=0THEN830ELSEIFI=100THEN850ELSE790
810 C2=VAL(A$):IFC2<1OR C2>6THEN820ELSE ON C2 GOTO 860,1350,1440,1130,1240,1470
820 GOSUB2260:LOCATE14,14:GOSUB2540:PRINTC$(1);"ｺﾏﾝﾄﾞ ﾊ ｲｶ ﾉ ﾄｵﾘﾃﾞｽ｡":LOCATE21,1
5:PRINT"ｵﾜｽﾚ ﾃﾞｽｶ ? ｾﾝﾁｮｳ ?":LOCATE14,17:PRINT"1:";F$(1);"     2:";F$(9):LOCATE14
,18:PRINT"3:ﾀﾞﾒｰｼﾞ･ﾚﾎﾟｰﾄ    4:";F$(4):LOCATE14,19:PRINT"5:";F$(5);"     6:";F$(10
):GOTO760
830 GOSUB2260:LOCATE14,14:GOSUB2550:PRINTC$(2):LOCATE14,15:PRINT"ｶｰｸ｡ ｱﾏﾘ ｶﾝｶﾞｴ
ｽｷﾞﾙﾄ ｾｲｼﾝｴｲｾｲﾆ ﾜﾙｲｿﾞ｡":LOCATE14,16:PRINT"ｶﾝ ｦ ｼﾝｼﾞﾙ ｺﾄﾓ ﾀｲｾﾂ ﾀﾞ｡":GOSUB2310
840 LOCATE14,18:GOSUB2540:PRINTC$(1):LOCATE14,19:PRINT"ﾄﾞｸﾀｰ ﾗｼｲ ｲｹﾝ ﾃﾞｽﾈ｡ ﾃﾞﾓ ﾜ
ﾀｼ ﾊ､":LOCATE14,20:PRINT"ﾖｸ ｶﾝｶﾞｴﾃ ﾛﾝﾘﾃｷﾆ ｺｳﾄﾞｳ ｽﾍﾞｷﾀﾞﾄ ｵﾓｲﾏｽ｡":GOTO780
850 GOSUB2260:LOCATE14,14:IFC1=0THENGOSUB2540:PRINTC$(1):LOCATE14,15:PRINT"ﾜﾀｼ ｶ
ﾞ ﾐﾃﾓ､ ﾁｮｯﾄ ｶﾝｶﾞｴｽｷﾞ ﾃﾞｽﾈｪ｡":C1=1:GOTO770ELSEGOSUB2380:GOSUB2540:PRINTC$(1);"ｾﾝﾁ
ｮｳ! ｵｷﾃ ｲﾏｽｶ!?":GOTO770
860 J=1:GOSUB2260:LOCATE14,14:GOSUB2560:PRINTC$(3):IFWC>4THENGOSUB2390:GOSUB2400
:GOTO730ELSEIFCE(1)>0THENGOSUB2420:LOCATE14,16:PRINT"ｾｲｾﾞｲ WARPO.3 ﾏﾃﾞ ｼｶ ﾄﾍﾞﾏｾﾝ
｡":LOCATE14,17:GOSUB2320:GOSUB2340:PRINT"ﾖﾛｼｲﾃﾞｽｶ ?":GOSUB1600:IFA$="N"THENGOTO7
50
```

```
870 Y=15:GOSUB2290:GOSUB2430:GOSUB2440
880 GOSUB2330:LOCATE14,17:GOSUB2340:PRINT"ﾊﾔｻ ｦ ｼﾚｲ ｼﾃｸﾀﾞｻｲ｡ ":LOCATE7,24:IFCE(1
)>0OR K=1THEN900
890 INPUT"WARP(0.1-12)";A$:WL!=VAL(A$):GOSUB2340:IFWL!>12THENLOCATE14,17:PRINT"W
ARP";WL!;"ﾅﾝﾃ ﾑﾘﾃﾞｽﾖ｡":GOTO910 ELSE920
900 INPUT"WARP(0.1-0.3)";A$:WL!=VAL(A$):GOSUB2340:IFWL!>.3THENLOCATE14,17:PRINT"
ｺｼｮｳﾁｭｳ ﾄ ｲｯﾀｼﾞｬﾅｲﾃﾞｽｶ!":LOCATE14,18:PRINT"WARP0.3 ﾏﾃﾞ ﾃﾞｽﾖ｡" ELSE920
910 GOSUB2310:Y=17:GOSUB2280:GOTO880
920 IFWL!<.1THENLOCATE14,17:PRINT"ﾄﾌﾞ ﾂﾓﾘｶﾞ ｱﾙﾝﾃﾞｽｶ ?":GOSUB1600:IFA$="N"THEN750
ELSEGOTO880
930 E!=ENG!-146-WL!^3/2:IFE!<=100THENLOCATE14,17:PRINT"ﾓｼ ﾄﾝﾀﾞﾗ ｴﾈﾙｷﾞｰ ｶﾞ":LOCAT
E14,18:PRINTE!;"ﾆﾅﾘﾏｽﾖ｡":LOCATE14,19:PRINT"ｲｲﾝﾃﾞｽｶ ?":GOSUB1600:IFA$="N"THEN910E
LSE Y=17:GOSUB2280
940 LOCATE14,17:PRINT"WARP";WL!;" ﾘｮｳｶｲ ｼﾏｼﾀ｡":GOSUB2310:LOCATE14,18:GOSUB2340:P
RINTF$(1);" ｼﾄﾞｳ!":GOSUB2310
950 GOSUB2260:WC=WC+1:PTC=0:PHC=0:YEAR=YEAR-1:SD=SD+1:FORI=1TO10:CE(I)=CE(I)-1:N
EXTI:ENG!=E!:WSX=SXE:WSY=SYE
960 GOSUB2070:X=8*QXE+SXE:Y=8*QYE+SYE:X=X+8*WL!*DX!:Y=Y+8*WL!*DY!:QX2=X¥8:QY2=Y¥
8:SX2=X MOD8:SY2=Y MOD8:GOSUB2080
970 GOSUB2090:IFX2<0ORX2>7ORY2<0ORY2>7THEN1040
980 IFS$(X2,Y2)<>"*"THEN1000ELSEJ=7:GOSUB2480:GOSUB2540:PRINTC$(1);"ｾﾞﾝﾎﾟｳ ﾌﾁｭｳｲ
 ﾄﾊ ﾋﾛﾝﾘﾃｷﾅ ｺｳﾄﾞｳ":LOCATE21,18:PRINT"ﾃﾞｽﾈ｡ ﾊﾞﾙｶﾝｼﾞﾝ ﾆﾊ ﾘｶｲ ﾃﾞｷﾏｾﾝ｡":IFCE(2)>=0T
HENLOCATE21,19:PRINTF$(2);"ｶﾞ ｺｼｮｳﾁｭｳ ﾅﾉﾃﾞ､":LOCATE21,20:PRINT"ﾑﾘﾓ ｱﾘﾏｾﾝｶﾞ｡"
990 GOSUB2310:GOSUB460
1000 IFS$(X2,Y2)="B"THENJ=8:GOSUB2480:GOSUB2540:PRINTC$(1);"ｷｺｳ ﾉ ｼｭﾀﾞﾝ ﾄｼﾃﾊ ﾔﾔ
ｶｹﾞｷ ﾃﾞｽﾈ｡":GOSUB2310:GOTO460
1010 IFS$(X2,Y2)="K"THENJ=9:GOSUB2480:GOSUB2540:PRINTC$(1);"ｶﾐｶｾﾞ ﾄﾊ ﾋﾛﾝﾘﾃｷﾅ ｺｳｲ
 ﾃﾞｽﾈ｡":GOSUB2310:GOTO460
1020 SQ$="E":C1=6:GOSUB2150:GOSUB2170
1030 SXE=X2:SYE=Y2:IF(SXE=SX2 AND SYE=SY2 AND QXE=QX2 AND QYE=QY2)OR SQR((WSX-X2
)*(WSX-X2)+(WSY-Y2)*(WSY-Y2))>8*WL! THEN460ELSE970
1040 GOSUB2150:GOSUB2310
1050 IFQX2<0ORQX2>7ORQY2<0ORQY2>7ORSX2<0ORSY2<0THENLOCATE14,14:GOSUB2570:PRINTC$
(4);"ｷﾞﾝｶﾞｹｲ ﾉ ｿﾄﾆ ﾃﾞﾃｼﾏｳ ﾄｺﾛﾃﾞｼﾀ｡":LOCATE21,15:PRINT"ﾃﾞﾓ､ ｺﾝﾋﾟｭｰﾀ ﾉ ｵｶｹﾞﾃﾞ ﾀｽｶﾘ
ﾏｼﾀ｡":LOCATE21,16:PRINT"ｼｶｼ､ ﾖﾃｲｶﾞｲﾉ QUADRANT ﾆ ﾂｷﾏｼﾀ｡"ELSE1070
1060 GOSUB2300:QX2=X:QY2=Y:GOSUB2300:SX2=X:SY2=Y
1070 IFCE(6)<0ANDCE(7)<0THENCL=7:X=QXE:Y=QYE:GOSUB1700:GOTO1100
1080 FORY=QYE-1TOQYE+1:FORX=QXE-1TOQXE+1:IFX<0ORX>7ORY<0ORY>7THEN1090ELSEGOSUB16
90
1090 NEXTX,Y
1100 QXE=QX2:QYE=QY2:SXE=SX2:SYE=SY2:SC=0
1110 IF(WL!>1.5AND RND>.85)OR(WL!>4.5AND RND>.75)THENGOSUB2310:GOSUB2260:LOCATE1
4,14:GOSUB2380:GOSUB2580:PRINTC$(5);"ｳﾁｭｳ ｱﾗｼ ﾊｯｾｲ!!":GOSUB2310:GOSUB1870
1120 GOTO370
1130 J=4:GOSUB2260:LOCATE14,14:GOSUB2580:PRINTC$(5):IFCE(4)>0THENGOSUB2420:GOTO7
40ELSE IFPHC>3THENGOSUB2390:GOSUB2410:GOTO730
1140 GOSUB2430:LOCATE14,16:GOSUB2320:GOSUB2340:PRINT"ｴﾈﾙｷﾞｰ ｦ ｼﾚｲ ｼﾃｸﾀﾞｻｲ｡":LOCA
TE7,24:INPUT"ENERGY";A$:E!=VAL(A$):LOCATE14,16:GOSUB2340
1150 IFE!>ENG!THENPRINT"ｴﾈﾙｷﾞｰ ｶﾞ ﾀﾘﾏｾﾝ｡      ":GOSUB2310:GOTO1140
1160 IFE!<=0THENPRINT"ﾊｯｼｬ ﾄﾘﾔﾒ ﾃﾞｽｶ ?    ":GOSUB1600:IFA$="N"THEN1140ELSE750
1170 IFENG!-E!<=50AND KLG-SK>0AND SB=0AND PT<5THENLOCATE14,18:GOSUB2540:PRINTC$(
1):LOCATE14,19:PRINT"ｿﾝﾅﾆ ｴﾈﾙｷﾞｰ ｦ ﾂｶｳﾄ ｱﾄﾃﾞ ｺﾏﾘﾏｽ｡":LOCATE14,20:PRINT"ﾛﾝﾘﾃｷﾆ ｶﾝ
ｶﾞｴﾃ ﾍﾝｺｳ ｽﾙﾍﾞｷﾃﾞｽ｡";:GOSUB2310:GOSUB2310:Y=16:GOSUB2290ELSE1190
1180 LOCATE14,16:GOSUB2580:PRINT"ﾍﾝｺｳ ｼﾏｽｶ ?":GOSUB1600:IFA$="N"THENLOCATE14,16E
LSE1140
1190 PRINT"ENERGY";E!;"ﾘｮｳｶｲ! ";F$(4);"ﾊｯｼｬ!":GOSUB2310
1200 Y=14:GOSUB2290:WC=0:PTC=0:PHC=PHC+1:ENG!=ENG!-E!:LOCATE15,14:PRINTC$(5):IFS
K=0THENLOCATE14,15:GOSUB2360:PRINT"KLINGON ﾊ ｲﾏｾﾝ ﾑﾀﾞｳﾁ ﾃﾞｼﾀ｡":GOTO470ELSE E!=E!
/SK:FORI=1TO5:GOSUB2120:IFK(I)=0THEN1230ELSE KE!(I)=KE!(I)-E!*FNL!(X1,Y1,X2,Y2)
1210 IFKE!(I)<=0THENGOSUB2130:SQ$="･":C1=3:A$="ｦ ﾊｶｲ!!"ELSE SQ$="K":C1=2:A$="ﾆ ﾀ
ﾞﾒｰｼﾞ!"
1220 LOCATE14,CSRLIN:COLOR6:PRINTUSING"#-# ﾉ KLINGON &        &";X2+1,Y2+1,A$:GOS
UB2180:IFCE(2)>0THENGOSUB2340
1230 NEXTI:COLOR7:GOTO470
1240 J=5:GOSUB2260:LOCATE14,14:GOSUB2570:PRINTC$(4):IFCE(5)>0THENGOSUB2420:GOTO7
40ELSE IFPTC>4THENGOSUB2390:GOSUB2410:GOTO730
1250 IFPT<=0THENLOCATE14,15:PRINTF$(5);"ﾊ ｱﾘﾏｾﾝ｡":GOTO740ELSEGOSUB2430:GOSUB2440
:LOCATE14,17:PRINTF$(5);"ﾊｯｼｬ!":GOSUB2310
1260 Y=14:GOSUB2290:WC=0:PHC=0:PTC=PTC+1:PT=PT-1:GOSUB2070:GOSUB2080
1270 LOCATE14,14:GOSUB2090:IFX2<0ORX2>7ORY2<0ORY2>7THENGOSUB2360:PRINTC$(4);"ﾊｽﾞ
ﾚﾏｼﾀ｡":GOSUB2150:GOTO470ELSEGOSUB2140:C1=5:SQ$="◆":GOSUB2180
1280 C2=0:A$="ﾊｶｲ ｼ":SQ$="･":C1=3:IFS$(X2,Y2)<>"K"THEN1320ELSE I=9:IFRND<.1THEN
A$="ﾀﾞﾒｰｼﾞ ｦ ｱﾀｴ":SQ$="K":C1=2:C2=1
1290 GOSUB2490
1300 FORI=1TO5:IFX2=K(I)¥10-1AND Y2=K(I)MOD10-1THEN IFC2=0THENGOSUB2130ELSE KE!(
I)=KE!(I)/2
1310 NEXTI:GOSUB2180:GOTO470
1320 IFS$(X2,Y2)<>"*"THEN1340ELSE I=7:IFRND<.3THEN A$="ｼﾝｾｲ ｶﾞ ﾊｯｾｲ ｼ":SQ$="*":C
1=7:C2=1
1330 GOSUB2490:GOSUB2180:IFC2=1THENGOSUB2310:GOSUB2550:LOCATE14,18:PRINTC$(2):LO
CATE14,19:PRINT"ﾎｳｼｬﾉｳ ﾀｲﾘｮｳ ﾊｯｾｲ!":LOCATE14,20:PRINT"ｶﾗﾀﾞ ﾆ ﾖｸﾅｲｿﾞ! ｶｰｸ｡":GOTO4
70ELSE SS=SS-1:Q(QXE,QYE)=Q(QXE,QYE)-1:GOTO470
1340 IFS$(X2,Y2)<>"B"THEN1270ELSE I=8:GOSUB2490:GOSUB2180:LOCATE14,17:PRINT"ﾁｷｭｳ
ｼﾞﾝ ﾉ ｺｳｲ ﾊ ﾘｶｲﾆ ｸﾙｼﾐﾏｽ｡":GOSUB2310:LOCATE14,19:GOSUB2550:PRINTC$(2):LOCATE14,20
:PRINT"ｺﾚﾊ ｸﾞﾝﾎﾟｳ ｶｲｷﾞ ﾓﾉ ﾀﾞｿﾞ｡":SB=0:Q(QXE,QYE)=Q(QXE,QYE)-10:MB=MB+1:GOTO470
1350 J=9:GOSUB2260:LOCATE14,14:GOSUB2540:PRINTC$(1):IFCE(9)>0THENGOSUB2420:GOTO7
```

BASICプログラム・リスト

```
40ELSEGOSUB2200:LOCATE14,15:PRINT"ｾﾝﾁｮｳ､ ";F$(9);"ﾉ ｾｲﾄﾞﾊ ｱﾏﾘ ﾖｸｱﾘﾏｾﾝ｡":LOCATE14
,16:PRINT"ﾁｭｳｲ ｼﾃｸﾀﾞｻｲ｡":GOSUB2310:GOSUB2310:Y=15:GOSUB2290
1360 GOSUB2340:LOCATE14,15:PRINTF$(9);"ｻﾄﾞｳ ｼﾏｼﾀ｡"
1370 LOCATE33,15:C1=33:FORX=0TO7:FORY=0TO7
1380 IFS$(X,Y)<>"K"OR RND>.9THEN1410
1390 FORI=1TO5:GOSUB2120:IFX=X2 AND Y=Y2 THEN E!=KE!(I)
1400 NEXTI:GOSUB2210:GOSUB2230:GOTO1420
1410 IFS$(X,Y)="B"OR(S$(X,Y)="*"ANDRND<.2)THEN E!=150+RND*170:GOSUB2210:GOSUB223
0
1420 NEXTY,X:IFCSRLIN=15AND POS(0)=33THENLOCATE14,16:GOSUB2340:PRINT"ｾｲﾒｲ ﾊﾝﾉｳ ﾊ
 ｱﾘﾏｾﾝ｡"
1430 ENG!=ENG!-50:GOSUB2310:GOSUB2310:GOTO470
1440 GOSUB2260:LOCATE14,14:GOSUB2360:PRINT"ｺﾝﾋﾟｭｰﾀ : ｾﾝﾅｲ ｼﾞｮｳｷｮｳ ｦ ｵｼﾗｾｼﾏｽ｡":GO
SUB2320
1450 LOCATE40,15:C1=40:FORI=1TO10:GOSUB2210:GOSUB2340:PRINTUSING"&          &: "
;F$(I);:IFCE(I)>0THENGOSUB2240ELSEGOSUB2250
1460 NEXTI:GOTO730
1470 J=10:GOSUB2260:LOCATE14,14:GOSUB2560:PRINTC$(3):IFCE(10)>0THENGOSUB2420:GOT
O740ELSEIF PT=0THENLOCATE14,15:PRINTF$(5);"ｶﾞ ｱﾘﾏｾﾝ｡":GOTO740ELSEGOSUB2200:GOSUB
2430
1480 LOCATE14,16:GOSUB2340:PRINT"ﾅﾝｺ ﾉ ";F$(5);"ｦ ｴﾈﾙｷﾞｰ ﾆ ﾍﾝｶﾝ ｼﾏｽｶ ?":LOCATE14
,17:PRINT"1ｺ ﾆ ﾂｷ 300 UNITS ﾃﾞｽ｡":LOCATE7,24:INPUT"HOW MANY";A$:C1=VAL(A$)
1490 LOCATE14,18:GOSUB2340:Y=18:IF PT<C1 THENPRINTF$(5);"ｶﾞ ﾀﾘﾏｾﾝ｡":GOSUB2310:GO
SUB2290:GOTO1480
1500 IFC1<=0THENPRINT"ﾄﾘﾔﾒ ﾃﾞｽｶ ?":GOSUB1600:IFA$="N"THENGOSUB2290:GOTO1480ELSE7
50
1510 PRINTC1;"ｺ ﾘｮｳｶｲ｡";:GOSUB2310:GOSUB2340:PRINT" ﾍﾝｶﾝ ｶﾝﾘｮｳ ｼﾏｼﾀ｡":ENG!=ENG!+
300*C1:PT=PT-C1:GOTO470
1520 GOSUB2260:LOCATE14,14:GOSUB2540:PRINTC$(1)+A$:LOCATE21,15:PRINT"ﾜﾚﾜﾚ ﾉ ﾏｹﾃﾞ
ｽ｡":GOSUB2310:LOCATE14,17:GOSUB2550:PRINTC$(2);"ｸﾖｸﾖ ｽﾙﾅ｡ ｶｰｸ｡":GOSUB2310:GOSUB2
510
1530 GOSUB2520:PRINT"ｼｯﾊﾟｲ ﾊ ﾜｶﾞﾎｳ ﾆ ﾀﾀﾞｲﾅ ｿﾝｶﾞｲ ｦ ﾓﾀﾗｼﾀ｡":LOCATE14,16:PRINT"ﾖｯﾃ
 ｷﾐ ﾆ ﾁｼﾞｮｳ ｷﾝﾑ ｦ ﾒｲｽﾞﾙ｡ ﾏｧ ｼｶｼ､":GOTO1590
1540 GOSUB2260:LOCATE14,14:GOSUB2540:PRINTC$(1);C$(9);"ﾊ ｾﾞﾝﾒﾂ ｼﾏｼﾀ｡":GOSUB2310:
LOCATE14,16:GOSUB2550:PRINTC$(2);"ﾖｸﾔｯﾀｿﾞ! ｶｰｸ!":GOSUB2310:GOSUB2310:GOSUB2510
1550 GOSUB2520:PRINT"ｶﾂﾔｸ ﾆ ﾀｲｼﾃ､ ﾋｮｳｶ ｦ ｱﾀｴﾙ｡":GOSUB2320:YR=YR-YEAR:LOCATE15,17
:GOSUB2340:PRINTKL;"ｾｷ ﾉ ";C$(9);"ｦ";YR;"ﾈﾝ ﾃﾞ ﾊｶｲ｡":LOCATE15,18:IFDE=0THEN1560E
LSEGOSUB2340:PRINTDE;"ﾒｲ ｾﾝｼ｡ ";
1560 IFMB=0THENPRINT:GOTO1570ELSEGOSUB2340:PRINTMB;"ｺﾉ ｷﾁ ｦ ﾊｶｲ｡"
1570 C1=500*(1+KL/YR-MB-DE/5):IFC1>1000THEN A$="A"ELSEIFC1>750THEN A$="B"ELSEIFC
1>500THEN A$="C"ELSEA$="D"
1580 LOCATE15,CSRLIN:GOSUB2340:PRINT" SCORE:";C1;"   CLASS: ";A$
1590 SC=0:LOCATE14,CSRLIN+1:GOSUB2340:PRINT"ﾓｳ 1ｶｲ ﾔﾙｶﾈ ?":GOSUB1600:IFA$="N"THE
NGOSUB2260:LOCATE14,14:GOSUB2380:PRINT"GOOD BYE!!":GOSUB2310:CONSOLE0,25:STOP:GO
TO120ELSEGOSUB1640:GOSUB1680:GOTO150
1600 GOSUB2320:GOSUB2340
1610 LOCATE7,24:PRINT"ANSWER(Y/N) ?";:FORT=0TO200:NEXTT:LOCATE7,24:PRINTSPC(13);
:A$=INKEY$:IFA$="Y"ORA$="N"THENRETURNELSEIFA$="y"THEN A$="Y":RETURNELSEIFA$="n"T
HEN A$="N":RETURNELSE1610
1620 FORY=0TO7:FORX=0TO7:IFQS(X,Y)=0THEN SQ$="***"ELSEGOSUB1650
1630 SYMBOL(FNMX(X,Y),FNSY(Y)),"■■■",1,1,0:SYMBOL(FNMX(X,Y),FNSY(Y)),SQ$,1,1,7:N
EXTX:NEXTY:RETURN
1640 FORY=0TO7:FORX=0TO7:SYMBOL(FNMX(X,Y),FNSY(Y)),"■■■",1,1,0:NEXTX,Y:RETURN
1650 IFQS(X,Y)=0THEN SQ$="***":RETURN
1660 IFQD(X,Y)>99THEN SQ$=RIGHT$(STR$(QD(X,Y)),3)ELSEIFQD(X,Y)>9THEN SQ$="0"+RIG
HT$(STR$(QD(X,Y)),2)ELSE SQ$="00"+RIGHT$(STR$(QD(X,Y)),1)
1670 RETURN
1680 FORY=0TO7:FORX=0TO7:SYMBOL(FNSX(X,Y),FNSY(Y)),"■",1,1,0:NEXTX,Y:RETURN
1690 SYMBOL(FNMX(X,Y),FNSY(Y)),"■■■",1,1,0:RETURN
1700 GOSUB1650:SYMBOL(FNMX(X,Y),FNSY(Y)),SQ$,1,1,CL:RETURN
1710 FORX=SXE-1TOSXE+1:FORY=SYE-1TOSYE+1:IFX<0ORX>7ORY<0ORY>7THEN1740ELSEIFS$(X,
Y)="B"THENCE(0)=5:ENG!=4000:PT=10:WC=0:PTC=0:PHC=0ELSE1740
1720 FORI=1TO10:IFCE(I)>0THENCE(I)=0
1730 NEXTI
1740 NEXTY,X:RETURN
1750 LOCATE26,2:PRINTSD:LOCATE28,5:PRINTYEAR:SYMBOL(16,95),"■",2,1,0:SYMBOL(39,9
4),"■",1,1,0:SYMBOL(593,94),"■",1,1,0:SYMBOL(608,95),"■",2,1,0:SYMBOL(16,95),RIG
HT$(STR$(QXE+1),1),2,1,7:SYMBOL(39,94),RIGHT$(STR$(QYE+1),1),1,1,7
1760 SYMBOL(593,94),RIGHT$(STR$(SXE+1),1),1,1,7:SYMBOL(608,95),RIGHT$(STR$(SYE+1
),1),2,1,7
1770 CE(0)=4:GOSUB1710
1780 IFCE(0)=5THEN1820
1790 GOSUB1850
1800 IFSK>2ORDM>4OR(CE(8)>0ANDSK>0)ORYEAR<2ORENG!<80THENCE(0)=2:GOTO1820
1810 IFSK>0ORDM>2ORCE(8)>0ORYEAR<4ORENG!<300THENCE(0)=6
1820 LOCATE18,12:PRINT USING"####.##";ENG!:LOCATE39,12:PRINT USING"##";PT:LOCATE
52,12:PRINT USING"##";KLG
1830 LOCATE66,12:IFCE(0)=5THENCOLOR5:PRINT"*D*"ELSECOLOR CE(0):PRINT"■■■"
1840 COLOR7:IFENG!<=0THEN A$=F$(0)+"ﾉ ｴﾈﾙｷﾞｰ ｶﾞ ｷﾚﾏｼﾀ｡":RETURN1520 ELSEIFYEAR<0T
HEN A$="TIME OVER ﾃﾞｽ｡":RETURN1520 ELSEIFKLG=0THENRETURN1540 ELSERETURN
1850 DM=0:FORI=1TO10:IFCE(I)>0THENDM=DM+1
1860 NEXTI:RETURN
1870 RANDOMIZETIME-INT(TIME/28800)*28800:I=INT(RND*10)+1:IFCE(I)>0THEN1870ELSEIF
(I=7ORI=8)ANDRND<=.2THEN1870
1880 CE(I)=INT(RND*5)+1
1890 GOSUB2260:LOCATE14,15:GOSUB2380:GOSUB2540:PRINTC$(1);F$(I);"ｶﾞ ﾀﾞﾒｰｼﾞｦ ｳｹﾏｼ
```

```
ﾀ｡":LOCATE20,16:PRINT"ｼｭｳﾘﾆ";CE(I);"ﾈﾝ ｶｶﾘﾏｽ｡"
1900 IFI=7THENLOCATE20,17:PRINT"MAP ﾉ ﾃﾞｰﾀ ｶﾞ ｼｮｳﾒﾂ ｼﾏｼﾀ｡"
1910 IFI=8THENLOCATE20,17:PRINT"ｸﾘﾝｺﾞﾝ ﾉ ｺｳｹﾞｷﾆ ﾀｲｼﾃ ﾑﾎﾞｳﾋﾞﾃﾞｽ｡"
1920 IFI=2THENGOSUB1680ELSEIFI=6THENGOSUB1640ELSEIFI=7THENGOSUB1640:FORX=0TO7:FO
RY=0TO7:QS(X,Y)=0:NEXTY,X
1930 RETURN
1940 I=INT(RND*11)-5:IFI<1THENRETURNELSEGOSUB2260:LOCATE14,15:GOSUB2380:GOSUB255
0:PRINTC$(2);"ﾃﾞｯｷ ｶﾞ ﾔﾗﾚﾃ､";I;"ﾒｲ ｼﾝﾀﾞ｡":LOCATE20,16:PRINT"ﾅﾆｦ ﾔｯﾃｲﾙﾝﾀﾞ! ｶｰｸ｡"
1950 DE=DE+I:RETURN
1960 E!=1:FORI=1TO10:E!=E!*CE(I):NEXTI:IFE!<>0THEN2030
1970 GOSUB2260:LOCATE14,14:GOSUB2540:PRINTC$(1)
1980 C1=-1:FORI=1TO10:IFCE(I)=0THENC1=(C1+1)MOD7ELSE2000
1990 LOCATE14,15+C1:GOSUB2340:PRINTF$(I);"ﾉ ｼｭｳﾘ ｶﾝﾘｮｳ ｼﾏｼﾀ｡":IFCSRLIN=22THENGOS
UB2310:Y=15:GOSUB2290
2000 NEXTI:GOSUB2050
2010 FORI=1TO10:IFCE(I)=0THENCE(I)=-1
2020 NEXTI:RETURN
2030 I=INT(RND*10)+1:IFCE(I)>0AND RND>.6THENGOSUB2310:GOSUB2260:LOCATE14,14:GOSU
B2540:PRINTC$(1);"ｱﾀﾗｼｲ ﾃｸﾆｯｸ ｦ ﾂｶｯﾃ､":LOCATE21,15:PRINTF$(I);"ｦ ｼｭｳﾘ ｼﾏｼﾀ｡":CE(
I)=0:GOSUB2050:CE(I)=-1
2040 RETURN
2050 IF(CE(6)=0ANDCE(7)<=0)OR(CE(6)<=0ANDCE(7)=0)THENGOSUB1620
2060 IFCE(2)=0THEN SC=0:RETURNELSERETURN
2070 CS!=CS!-1:DX!=COS(CS!*.785398):DY!=-SIN(CS!*.785398):RETURN
2080 X1=SXE:Y1=SYE:X2=SXE:Y2=SYE:X3!=SXE:Y3!=SYE:RETURN
2090 X3!=X3!+DX!:Y3!=Y3!+DY!:X1=X2:Y1=Y2:X2=X3!:Y2=Y3!:RETURN
2100 SYMBOL(FNSX(X,Y),FNSY(Y)),"■",1,1,0
2110 SYMBOL(FNSX(X,Y),FNSY(Y)),A$,1,1,CL:RETURN
2120 X1=SXE:Y1=SYE:X2=K(I)¥10-1:Y2=K(I)MOD10-1:RETURN
2130 K(I)=0:SK=SK-1:KLG=KLG-1:S$(X2,Y2)="･":Q(QXE,QYE)=Q(QXE,QYE)-100:RETURN
2140 IFS$(X1,Y1)="E"THENRETURN
2150 S$(X1,Y1)="･":IFCE(2)<0THEN X=X1:Y=Y1:CL=3:A$="･":GOSUB2100
2160 RETURN
2170 S$(X2,Y2)=SQ$
2180 IFCE(2)<0THEN X=X2:Y=Y2:CL=C1:A$=SQ$:GOSUB2100:GOSUB2340
2190 RETURN
2200 WC=0:PHC=0:PTC=0:RETURN
2210 IFPOS(0)<C1 THENLOCATEC1,CSRLIN ELSELOCATE14,CSRLIN+1
2220 RETURN
2230 GOSUB2340:PRINTUSING"#-#: ###.##UNITS";X+1,Y+1,E!;:RETURN
2240 PRINTUSING"ﾀﾞﾒｰｼﾞ(#ﾈﾝ)";CE(I);:RETURN
2250 PRINT"Ok";:RETURN
2260 LINE(112,112)-(527,175),PSET,0,BF:RETURN
2270 LINE(FX1,FY1)-(FX2,FY2),PSET,0,BF:RETURN
2280 LINE(112,8*Y)-(319,175),PSET,0,BF:RETURN
2290 LINE(112,8*Y)-(415,175),PSET,0,BF:RETURN
2300 X=INT(RND*8):Y=INT(RND*8):RETURN
2310 FORT=0TO700:NEXTT:RETURN
2320 FORT=0TO150:NEXTT:RETURN
2330 FORT=0TO50:NEXTT:RETURN
2340 IFO=1THENBEEP1:FORT=0TO5:NEXTT:BEEP0
2350 RETURN
2360 IFO=1THENBEEP1:FORT=0TO50:NEXTT:BEEP0
2370 RETURN
2380 FORK=1TO2:GOSUB2360:GOSUB2330:NEXTK:RETURN
2390 LOCATE14,15:PRINT"ｾﾝﾁｮｳ､ ";F$(J);"ﾊ ｵｰﾊﾞｰﾛｰﾄﾞ ﾃﾞｽ｡":RETURN
2400 IFCE(4)>0ANDCE(5)>0ANDCE(9)>0ANDCE(10)>0THENGOSUB2320:GOSUB2340:LOCATE14,16
:PRINT"ﾃﾞﾓ ﾎｶﾉ ｺﾏﾝﾄﾞ ｶﾞ ﾐﾝﾅ ｺｼｮｳ ｼﾃｲﾙﾉﾃﾞ､":LOCATE14,17:PRINT"ｳｺﾞｶｼ ﾏｼｮｳ｡ ﾀﾀﾞｼ､
WARP0.3 ﾏﾃﾞﾃﾞｽ｡":K=1:GOSUB2310:GOSUB2310:RETURN870ELSE K=0
2410 LOCATE14,16:PRINT"ｲﾁﾄﾞ ﾍﾞﾂﾉ ｺﾏﾝﾄﾞ ｦ ｼﾚｲ ｼﾃ ｸﾀﾞｻｲ｡":LOCATE14,17:PRINT"(ｺﾏﾝﾄﾞ
3:ﾀﾞﾒｰｼﾞ･ﾚﾎﾟｰﾄ ﾊ ﾉｿﾞｸ)":RETURN
2420 LOCATE14,15:PRINT"ｾﾝﾁｮｳ､ ";F$(J);"ﾊ ｺｼｮｳ ｼﾃｲﾏｽ｡":RETURN
2430 LOCATE14,15:PRINTF$(J);"ｼﾞｭﾝﾋﾞ ｶﾝﾘｮｳ｡":RETURN
2440 LINE(328,156)-(383,156),PSET,7:LINE(356,144)-(356,167),PSET,7:CIRCLE(356,15
6),20,3,.4:CIRCLE(356,156),12,6,.4:LOCATE48,19:PRINT"1":LOCATE44,17:PRINT"3":LOC
ATE40,19:PRINT"5":LOCATE44,21:PRINT"7"
2450 LOCATE14,16:GOSUB2340:PRINT"ｺｰｽ ｦ ｼﾚｲ ｼﾃｸﾀﾞｻｲ｡":LOCATE7,24:INPUT"COURSE(1-8
.9)";A$:IFA$="0"THEN CS!=.1ELSECS!=VAL(A$)
2460 LOCATE14,16:GOSUB2340:IFCS!=0THENPRINT"ｺﾏﾝﾄﾞ ﾄﾘｹｼ ﾃﾞｽｶ ? ":GOSUB1600:IFA$="
N"THEN2450ELSERETURN750
2470 IFCS!<1ORCS!>8.9THENPRINT"1-8.9 ﾉ ﾊﾝｲ ﾃﾞｽﾖ｡ ":GOSUB2310:GOTO2450ELSEPRINT"C
OURSE";CS!;" ﾘｮｳｶｲ ｼﾏｼﾀ｡":RETURN
2480 SXE=X1:SYE=Y1:GOSUB2570:LOCATE14,14:PRINTC$(4);"ｷﾝｷｭｳ ﾃｲｼ!!":LOCATE21,15:PR
INT"ｾﾞﾝﾎﾟｳ ﾆ ";C$(J);"ｶﾞ ｱﾘﾏｽ｡":GOSUB2310:LOCATE14,17:RETURN
2490 GOSUB2540:PRINTC$(1):LOCATE14,15:PRINTC$(I);"ﾆ ﾒｲﾁｭｳ｡":LOCATE14,16:PRINTA$;
"ﾏｼﾀ｡":IFC2=0THEN S$(X2,Y2)="･"
2500 RETURN
2510 GOSUB2260:GOSUB2590:LOCATE14,14:PRINTC$(6);"ｳﾁｭｳ ﾚﾝﾎﾟｳ ﾖﾘ ﾆｭｳﾃﾞﾝ｡":LOCATE21
,15:PRINT"ﾁｬﾝﾈﾙ ｵｰﾌﾟﾝ ｼﾏｽ｡":GOSUB2310:RETURN
2520 GOSUB2260:GOSUB2360:LOCATE15,14:PRINT"ｶｰｸ ｾﾝﾁｮｳ ﾆ ﾂｸﾞ｡":LOCATE14,15:PRINT"ｺ
ﾝｶｲ ﾉ ｷﾐ ﾉ ";:RETURN
2530 AD1=&H6F:AD2=&H98:CL=6:GOSUB 2592:RETURN
2540 AD1=&H71:AD2=&HF0:CL=5:GOSUB 2592:RETURN
2550 AD1=&H74:AD2=&H48:CL=5:GOSUB 2592:RETURN
2560 AD1=&H76:AD2=&HA0:CL=2:GOSUB 2592:RETURN
2570 AD1=&H78:AD2=&HF8:CL=6:GOSUB 2592:RETURN
```

BASICプログラム・リスト

```
2580 AD1=&H7B:AD2=&H50:CL=2:GOSUB 2592:RETURN
2590 AD1=&H7D:AD2=&HA8:CL=2:GOSUB 2592:RETURN
2592 POKE&H6F86,AD1:POKE &H6F87,AD2:POKE &H6F8D,(VARPTR(FA(0)) ¥ 256):POKE &H6F8
E,(VARPTR(FA(0)) MOD 256)
2594 EXEC &H6F85:GOSUB2360:GOSUB2270:PUT@(FX1,FY1)-(FX2,FY2),FA,PSET,CL:RETURN
2600 CLS:LINE(0,0)-(639,199),PSET,7,BF
2610 FORI=-1TO1STEP2
2620 CONNECT(FNX(320,I),0)-(FNX(251,I),87)-(320,87),0:CONNECT(FNX(320,I),15)-(FN
X(251,I),92)-(FNX(251,I),90)-(320,90),0:CONNECT(FNX(251,I),87)-(FNX(251,I),104)-
(FNX(256,I),111)-(FNX(286,I),135)-(FNX(319,I),140),0
2630 CONNECT(FNX(286,I),135)-(FNX(251,I),172)-(FNX(251,I),182),0:CONNECT(FNX(258
,I),110)-(FNX(319,I),116),0
2640 CONNECT(FNX(251,I),172)-(FNX(273,I),180),0:CONNECT(FNX(1,I),187)-(FNX(296,I
),187)-(FNX(304,I),188)-(FNX(320,I),192),1:CONNECT(FNX(1,I),182)-(FNX(300,I),182
)-(FNX(310,I),183)-(FNX(320,I),185),0
2650 CIRCLE(FNX(304,I),160),16,2,1:CIRCLE(FNX(296,I),160),16,0,1:PAINT(FNX(300,I
),140),5,0:CIRCLE(FNX(296,I),160),20,0,1:PAINT(FNX(279,I),160),2,0:PAINT(FNX(285
,I),160),1,0,2:PAINT(FNX(296,I),160),0,2
2660 CONNECT(FNX(294,I),153)-(FNX(298,I),150)-(FNX(304,I),172),6:CONNECT(FNX(307
,I),155)-(FNX(294,I),163),2
2670 CONNECT(FNX(306,I),148)-(FNX(301,I),155)-(FNX(299,I),162)-(FNX(297,I),169)-
(FNX(297,I),174),7:CONNECT(FNX(307,I),164)-(FNX(299,I),162)-(FNX(291,I),160),7
2680 CONNECT(FNX(304,I),51)-(FNX(297,I),51)-(FNX(268,I),87)-(FNX(265,I),93)-(FNX
(265,I),102)-(FNX(269,I),105)-(FNX(297,I),107)-(FNX(304,I),105)-(FNX(304,I),51),
0:CONNECT(FNX(304,I),53)-(FNX(274,I),87)-(FNX(271,I),93)-(FNX(271,I),102)-(FNX(2
76,I),105),0
2690 CONNECT(FNX(308,I),49)-(FNX(294,I),49)-(FNX(265,I),85)-(FNX(261,I),93)-(FNX
(261,I),102)-(FNX(266,I),107)-(FNX(300,I),109)-(FNX(308,I),107)-(FNX(308,I),49),
0:CONNECT(FNX(304,I),53)-(FNX(274,I),87)-(FNX(271,I),93)-(FNX(271,I),102)-(FNX(2
76,I),105),0
2700 CONNECT(FNX(320,I),26)-(FNX(315,I),31)-(FNX(313,I),35)-(FNX(313,I),105)-(FN
X(315,I),107)-(FNX(320,I),109),0:CONNECT(FNX(320,I),28)-(FNX(318,I),33)-(FNX(317
,I),35)-(FNX(317,I),105)-(FNX(318,I),105)-(FNX(320,I),107),0
2710 PAINT(FNX(267,I),87),2,0:PAINT(FNX(283,I),107),2,0:PAINT(FNX(316,I),70),2,0
:PAINT(FNX(271,I),87),1,0:PAINT(FNX(299,I),67),0:PAINT(FNX(320,I),70),1,0
2720 CONNECT(FNX(320,I),121)-(FNX(273,I),115)-(FNX(291,I),131)-(FNX(320,I),135),
0:LINE(FNX(283,I),123)-(FNX(320,I),128),PSET,0:LINE(FNX(293,I),117)-(FNX(304,I),
132),PSET,0:LINE(FNX(310,I),119)-(FNX(319,I),135),PSET,0
2730 PAINT(FNX(285,I),119),RND*5+1,0:PAINT(FNX(292,I),127),RND*5+1,0:PAINT(FNX(3
06,I),123),RND*5+1,0:PAINT(FNX(309,I),130),RND*5+1,0:PAINT(FNX(318,I),125),RND*5
+1,0:PAINT(FNX(320,I),131),RND*5+1,0
2740 CONNECT(FNX(1,I),111)-(FNX(213,I),111)-(FNX(220,I),119)-(FNX(220,I),167)-(F
NX(213,I),176)-(FNX(1,I),176),0:CONNECT(FNX(1,I),108)-(FNX(214,I),108)-(FNX(225,
I),119)-(FNX(225,I),167)-(FNX(214,I),179)-(FNX(1,I),179),0
2750 LINE(FNX(256,I),142)-(FNX(233,I),145),PSET,0,BF:LINE(FNX(252,I),150)-(FNX(2
33,I),153),PSET,0,BF:LINE(FNX(252,I),118)-(FNX(233,I),121),PSET,0,BF:LINE(FNX(25
6,I),126)-(FNX(233,I),129),PSET,0,BF:LINE(FNX(260,I),134)-(FNX(233,I),137),PSET,
0,BF
2760 LINE(FNX(288,I),98)-(FNX(284,I),98),PSET,7
2770 NEXTI
2780 CONNECT(24,7)-(149,7)-(155,11)-(183,76)-(183,79)-(77,79)-(73,76)-(24,11)-(2
4,7),0:CONNECT(32,8)-(82,75)-(182,75),0:CONNECT(19,4)-(155,4)-(159,9)-(187,78)-(
187,81)-(74,81)-(70,78)-(19,9)-(19,4),0:PAINT(67,61),5,0:PAINT(94,80),3,0
2790 CONNECT(291,7)-(618,7)-(618,11)-(565,76)-(561,79)-(295,79)-(293,76)-(288,11
)-(291,7),0:CONNECT(294,8)-(297,75)-(558,75)-(608,8),0:CONNECT(287,4)-(623,4)-(6
23,9)-(568,78)-(564,81)-(291,81)-(289,78)-(284,9)-(287,4),0:PAINT(330,77),5,0:PA
INT(320,80),3,0
2800 CONNECT(82,75)-(77,79),0:CONNECT(297,75)-(295,79),0:CONNECT(558,75)-(561,79
),0
2810 CONNECT(174,7)-(270,7)-(272,25)-(179,25)-(174,7),0:CONNECT(168,4)-(275,4)-(
277,27)-(176,27)-(168,4),0:CONNECT(183,31)-(271,31)-(273,48)-(188,48)-(183,31),0
:CONNECT(178,29)-(276,29)-(278,50)-(185,50)-(178,29),0:PAINT(200,6),6,0:PAINT(20
0,30),6,0
2820 CIRCLE(235,64),31,5,.3:CIRCLE(235,67),31,0,.3:PAINT(320,2),1,0:CIRCLE(235,6
7),39,0,.3:PAINT(241,76),3,0:PAINT(226,74),5,0,2:PAINT(235,65),0,5:CONNECT(247,6
0)-(225,70),4:CONNECT(216,67)-(216,63)-(255,67)-(255,63),6
2830 PAINT(320,94),5,0:PAINT(320,110),2,0:GOSUB2260:PAINT(110,116),0:PAINT(530,1
16),0:PAINT(100,9),0:PAINT(400,9),0:PAINT(200,8),0:PAINT(200,32),0:PAINT(320,188
),0,1
2840 LINE(84,94)-(203,105),PSET,6,BF:LINE(210,94)-(331,105),PSET,6,BF:LINE(340,9
4)-(435,105),PSET,6,BF:LINE(444,94)-(555,105),PSET,6,BF
2850 LINE(87,95)-(200,104),PSET,0,BF:LINE(215,95)-(328,104),PSET,0,BF:LINE(343,9
5)-(432,104),PSET,0,BF:LINE(447,95)-(552,104),PSET,0,BF
2860 SYMBOL(24,80),"Q",2,1,7:SYMBOL(600,80),"S",2,1,7:LOCATE23,1:PRINT"STAR DATE
":LOCATE24,4:PRINT"YEARS LEFT":LOCATE11,12:PRINT"ENERGY:":LOCATE27,12:PRINT"PHOT
ON TOR.:":LOCATE43,12:PRINT"KLINGONS:":LOCATE56,12:PRINT"INDICATOR"
2870 RETURN
2880 CLS:FOR I=1 TO 300:X=RND*638:Y=RND*198:CL=INT(RND*7+1):PSET(X,Y,CL):PSET(X+
1,Y,CL):NEXT
2890 CIRCLE(320,90),96,1,.21,.42,.35:CIRCLE(320,93),88,1,.19,.42,.36:CIRCLE(320,
98),12,1,.21:CIRCLE(320,96),40,5,.19:CONNECT(220,113)-(240,103)-(237,100)-(284,9
9)-(251,111),1
2900 CONNECT(278,114)-(274,114)-(264,111)-(165,115),1:CONNECT(155,115)-(130,116)
-(124,120)-(139,122)-(145,124)-(146,125)-(264,125)-(274,122)-(278,122),1:CONNECT
(278,117)-(287,117),1:CONNECT(137,98)-(157,118)-(169,118)-(150,98),1
2910 CONNECT(191,90)-(45,96)-(65,101)-(191,96),1:CONNECT(221,99)-(154,102),1:CON
NECT(142,102)-(110,103)-(130,108)-(146,107),1:CONNECT(159,107)-(181,107),1:CONNE
```

```
CT(190,106)-(221,105),1:CONNECT(176,114)-(182,105)-(192,105)-(186,114),1
2920 CIRCLE(278,118),4,1,1:CIRCLE(274,118),4,1,1,.25,.75:CIRCLE(264,118),7,1,1,.
25,.75:CIRCLE(183,93),3,1,1,.25,.75:CIRCLE(191,93),3,1,1,.25,.75:CIRCLE(191,93),
6,1,.5,.75,.25:CIRCLE(213,102),3,1,1,.25,.75:CIRCLE(221,102),3,1,1,.25,.75
2930 CIRCLE(221,102),6,1,.5,.75,.25
2940 PAINT(246,82),5,1:PAINT(249,89),7,1,5:PAINT(292,96),7,1,5:PAINT(320,98),5,1
:PAINT(251,106),7,1:PAINT(251,113),7,1:PAINT(264,116),0,1:PAINT(272,118),7,1:PAI
NT(137,95),7,1:PAINT(136,105),7,1:PAINT(169,104),7,1:PAINT(149,104),7,1
2950 PAINT(184,109),7,1:PAINT(193,93),2,1:PAINT(223,102),2,1:PAINT(266,118),3,1:
PAINT(278,118),3,1:PAINT(215,102),6,1:PAINT(186,93),6,1:CONNECT(245,116)-(182,11
9)-(245,120),2:CIRCLE(245,118),4,2,.5,.75,.25:PAINT(232,118),2,2
2960 X1=20:Y1=8:C1=4:FORI=0TO6:CL=3:A$=LEFT$(C$(I),6):GOSUB3020:FX1=90*I:FY1=138
:FX2=FX1+95:FY2=187:ONI+1GOSUB2530,2540,2550,2560,2570,2580,2590:GOSUB2320:CL=0:
GOSUB3020:NEXTI:GOSUB2320
2970 X1=112:Y1=15:C1=3:CL=7:A$="in":GOSUB3020:GOSUB2320:GOSUB2320:CL=0:GOSUB3020
:X1=20:Y1=8:C1=5:CL=2:A$="STAR TREK":GOSUB3020:X1=105:Y1=48:C1=2:CL=6:A$="THE CO
MPUTER GAME":GOSUB3020
2980 CONNECT(546,15)-(531,23)-(508,37)-(491,50)-(479,63)-(499,55)-(514,51)-(525,
49)-(536,48)-(538,48)-(545,50)-(552,55)-(560,62)-(560,48)-(557,32)-(552,22)-(548
,15)-(546,15),1:PAINT(545,20),5,1
2990 CONNECT(542,24)-(536,33)-(523,32)-(530,36)-(524,41)-(534,38)-(543,46)-(541,
36)-(554,34)-(542,34)-(542,24),1:PAINT(535,34),6,1
3000 LOCATE2,24:PRINT"Copyright (C) 1981 by M.Kasuya";
3010 RETURN
3020 SYMBOL(X1,Y1),A$,C1,C1,CL:RETURN
```

ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6F80 36 46 38 35 00 8E 00 00 10 8E 02 58 CE 00 00 EC :29
6F90 81 ED C1 31 3E 26 F8 39 FF FF FF FF FF FF FF FF :ED
6FA0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
6FB0 FF FF FF FF 00 00 00 7F FF FF FF FF FF FF FE 00 :73
6FC0 00 00 00 00 1F FF FF FF FF FF F0 04 82 00 00 00 :90
6FD0 00 FF FF FF FF FF E4 24 92 40 00 00 00 3F FF FF :12
6FE0 FF FF 00 24 92 40 20 00 04 07 FF FF FF FC 81 24 :BD
6FF0 90 00 92 80 05 47 FF FF FF F8 09 20 80 00 92 00 :1E
----------------------------------------------------------
Sum: 44 2F 88 07 F2 38 F9 D9 A1 C9 F7 78 CC 38 0E 0D :F6

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7000 15 4F FF FF FF F8 49 00 00 04 B7 3C 00 0F FF FF :A6
7010 FF E0 49 00 07 1F FF FF 80 03 FF FF FF E0 49 00 :F5
7020 2E 3F FF FF F4 01 FF FF FF 00 49 00 69 7F FF FF :8C
7030 FE 01 FF FF FF 00 48 01 63 FF FF FF FE 03 FF FF :A4
7040 FF 00 08 03 6F FF FF FF FC 01 FF FF FF 00 00 43 :B3
7050 6F FF FF FF FC 03 FF FF FF 00 41 5B 6F FF FF FF :70
7060 FE 03 FF FF FF 02 40 5B 67 FD FF F7 49 87 FF FF :C3
7070 FF 02 40 1B 64 90 3F E2 7F 0F FF FF FF 02 40 5B :99
7080 1B 00 0F 80 03 03 FF FF FF 00 40 59 70 10 07 86 :53
7090 4F 93 FF FF FF 00 41 5F C7 9B 97 8F FF D3 FF FF :D7
70A0 FF 80 01 5F FF FF CF CF FF C3 FF FF FF CE 01 5F :68
70B0 FF FF A7 E7 FF 83 FF FF FF D6 C1 6F FF ED 87 D3 :57
70C0 FF CF FF FF FF D6 11 6F FF EF 3C F9 FF CF FF FF :10
70D0 FF E6 59 6D FF FB 76 79 7F CF FF FF FF E2 59 6D :87
70E0 BF FB E1 B1 DF CF FF FF FF E0 1B 6D FF FB A8 DF :E0
70F0 CF CF FF FF FF F0 CB 6D FF CF FF FF E7 9F FF FF :13
----------------------------------------------------------
Sum: 9F 04 7A FA A3 C1 6B BA 03 B4 28 44 6D E2 11 9A :BD

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7100 FF FC 1A 6D FF 3F FF FF FF BF FF FF FF FF 88 6D :6D
7110 BF 30 00 00 7F 3F FF FF FF FF E4 6D BF B5 FF FE :6B
7120 FE FF FF FF FF FF F0 2D BF FF C8 87 F9 FF FF FF :19
7130 FF FF F9 0D B7 FF E0 3F F3 FF FF FF FF FF F9 45 :05
7140 B7 FF FF FF C7 FF FF FF FF FF F9 60 36 FF FF FD :00
7150 9F FF FF FF FF FF F1 6C 96 D9 03 26 3F FF FF FF :CB
7160 FF FF B1 6D A2 D3 69 98 7F FF FF FF FF F8 00 3F :14
7170 B0 1B 64 05 00 1F FF FF FF E1 80 07 FE 10 41 55 :5C
7180 00 C0 7F FF E0 0D 80 00 FE DB 4D 55 00 DB 01 FF :01
7190 03 6D 80 00 0F FF 45 FC 00 DB 6C 27 1B 6D 80 00 :B5
71A0 00 01 67 E0 00 DB 6D 91 DB 6D A0 00 00 00 00 00 :09
71B0 02 DB 6D B0 DB 6D B4 00 00 00 00 00 06 DB 6D B6 :FA
71C0 DB 6D B6 D0 00 00 00 00 06 DB 6D B6 DB 6D B6 DB :AB
71D0 00 00 00 00 B6 DB 6D B6 DB 6D B6 DB 6C 00 00 05 :FE
71E0 B6 DB 6D B6 DB 6D B6 DB 6D B6 DB 6D B6 DB 6D B6 :AC
71F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FE :EF
----------------------------------------------------------
Sum: 55 92 EA FD F6 07 2E 89 E9 94 7B F7 45 22 CE 88 :2E

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7200 00 00 00 07 FF FF FF FF FF FF FF C0 00 00 00 00 :C0
7210 3F FF FF FF FF FF FE 00 0E EE C0 00 07 FF FF FF :F8
7220 FF FF F8 00 E8 EE EE 00 01 FF FF FF FF FF E0 00 :96
7230 00 00 00 00 00 7F FF FF FF FF 80 00 00 00 00 00 :FB
7240 00 1F FF FF FF FF 00 00 00 00 00 03 A8 0F FF FF :D3
7250 FF FC 2A 00 00 00 00 0B B8 07 FF FF FF F0 3B 80 :97
7260 00 00 01 BA 08 03 FF FF FF E0 32 B8 00 00 3B 80 :48
7270 02 03 FF FF FF C0 A0 33 BB BB A0 00 02 03 FF FF :AE
7280 FF C1 80 00 00 00 00 00 02 01 FF FF FF C1 80 00 :81
7290 00 00 00 06 00 01 FF FF FF C1 00 00 00 00 05 B6 :80
72A0 00 01 BF FF FF C1 1F FF FF FF FD B6 00 01 3F FF :8D
72B0 FF B1 0F FF FF FF FD 90 80 00 BF FF FF 80 0C 7F :61
72C0 FF FF FE 40 C0 01 BF FF FF 80 0C 07 FF FF F8 1C :5F
72D0 C0 03 3F FF FF C0 1D E0 3F FF C1 FE D0 0E 3F FF :D6
72E0 FF C8 19 FF 01 FC 0F FF D8 1C 7F FF FF CC 1E 00 :45
72F0 00 F0 40 00 78 38 FF FF FF CE 1E 3C 3C F3 8C 3F :FF
----------------------------------------------------------
Sum: FB 19 04 00 22 E3 8E A6 14 B7 34 6D B7 0E 04 8B :11

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7300 F8 64 FF FF FF CB 1F FF FC FB FF FF F8 CC FF FF :F9
7310 FF E3 9F FF FC F9 FF FF F8 DC FF FF FF E4 CF FF :F6
7320 FC FD FF FF F0 B9 FF FF FF E4 CF FF F8 FF FF FF :44
7330 E0 B3 FF FF FF F3 C3 FF F8 FF 3F FF 89 E7 FF FF :E8
7340 FF F9 CC FF F0 FF 9F FF 29 C7 FF FF FF FC B6 BF :AE
7350 E6 FF CF FC A9 1F FF FF FF FE 07 BF E6 FF 9F FC :B9
7360 A8 FF FF FF FF FF F3 FF 88 18 3C FC A1 FF FF FF :0B
7370 FF FF F3 FE 3F FF FE 3C A3 FF FF FF FF FF F9 FC :FA
7380 FF FF FF 8C A3 FF FF FF FF FF FC FB FF FF FF E4 :FF
7390 A7 FF FF FF FF FF FE 7B 7F FF FB E4 8F FF FF FF :04
73A0 FF FF FF 3F 80 00 03 C8 07 FF FF FF FF FF FF 9F :27
73B0 FF FF FF 98 47 FF FF FF FF FF FF CF FC 00 7F B1 :D1
73C0 47 FF FF FF FF FF FF D7 FF FF FF C5 47 FF FF FF :1E
73D0 FF FF FF C8 FF FF FE 15 60 1F FF FF FF FF FF CE :1E
73E0 1F FF F8 55 80 00 7F FF FF FF FF 8F C0 00 01 5C :12
73F0 00 2D 80 1F FF FF FE 03 F5 45 57 F0 00 6D B6 00 :6F
----------------------------------------------------------
Sum: 68 13 9B 91 A7 86 E8 64 15 F4 96 A5 8C F7 4A 0E :3F

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7400 FF FF FC 00 7F FF FE 00 00 6D B6 DB FF FF 80 00 :F2
7410 00 00 00 00 03 6D B6 DB FF E0 58 00 00 00 00 00 :38
7420 0B 6D B6 DB E0 06 DB 00 00 00 00 00 1B 6D B6 DB :E3
7430 05 B6 DB 60 00 00 00 06 DB 6D B6 DB 6D B6 DB 68 :3B
7440 00 00 00 36 DB 6D B6 DB FF FF FF FF FF FF FF FF :07
7450 FF FF FF FF FF FF FF FF FC 00 00 00 7F FF FF FF :70
7460 FF FF FF FF 80 00 00 00 0F FF FF FF FF FF FF E0 :65
7470 00 02 A0 00 00 FF FF FF FF FF FF 00 00 00 00 00 :9C
7480 00 7F FF FF FF FF F8 00 00 AA A0 00 00 1F FF FF :DA
7490 FF FF C0 00 00 00 00 00 00 87 FF FF FF FE 00 00 :40
74A0 00 A0 00 00 00 C3 FF FF FF F8 00 00 00 00 00 00 :58
74B0 00 60 FF FF FF F0 00 80 00 00 00 00 00 30 7F FF :7B
74C0 FF E0 04 40 00 00 00 00 00 12 BF FF FF C0 C4 10 :86
74D0 00 00 00 02 00 32 3F FF FF C0 B2 02 40 00 00 A8 :9D
74E0 00 80 3F FF FF 80 C1 80 00 00 00 00 04 02 3F FF :C2
74F0 FF 00 60 78 00 00 00 00 00 02 3F FF FE 00 00 00 :15
----------------------------------------------------------
Sum: 0A 00 8C 26 B9 41 3A B8 E1 B4 E0 B3 44 2E 8F D6 :A7

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7500 00 00 0F F8 00 02 3F FF FE 00 00 00 00 03 FF C0 :07
7510 00 00 3F FF FE 00 F0 00 00 3F FF 00 00 00 3F FF :A8
7520 FF 01 FF FF FF FF FF FF 80 00 37 FF FF 81 FF FF :2E
7530 FF FF FF FE 10 00 3F FF FF 80 FF FF FF FF FF FF :C2
7540 F8 01 1F FF FF 80 FF 07 FF FF F8 1F F4 03 1F FF :C6
7550 FF 80 FC F0 FF FF C3 C7 F4 03 1F FF FF 80 F3 FF :79
7560 0F FE 1F F1 F6 03 1F FF FF 80 CF FF E1 F0 FF FF :50
7570 F6 07 3F FF FF 80 FE 00 05 F7 00 01 F6 8E 3F FF :77
7580 FF 98 FC C6 79 F9 E3 3C FF 86 7F FF FF 99 FF E0 :64
7590 F9 F9 E8 7F F1 8E 7F FF FF C9 FF FF F9 F8 FF FF :0B
75A0 C6 8C 7F FF FF CC FF E0 F9 F8 E0 3F C2 86 FF FF :D0
75B0 FF E4 FF FF F1 F8 3F FF CA A6 FF FF FF E7 1F FF :7A
75C0 E1 FD 1F FF CA 81 FF FF FF F1 27 FF E1 FD 1F FF :57
75D0 CA 3F FF FF FF FE 39 FF E7 F9 7F F5 CA 3F FF FF :97
75E0 FF FF 3F FF FE 07 FF F1 7A 7F FF FF FF FF 1F 7F :C4
75F0 FF FF FF F9 28 FF FF FF FF FF 8F 3F FD F7 FF FD :D7
----------------------------------------------------------
Sum: 60 C1 83 0C 49 D3 22 D2 94 8D AC 8A 28 B4 E4 10 :E7

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7600 29 3F FF FF FF FF D7 9F FF FF FF F9 20 1F FF FF :0D
7610 FF FF EB DF 80 00 0F F1 0A 0F FF FF FF FF F0 EE :3B
```

ダンプ・リスト

```
7620 3F FF F1 C4 0A 00 7F FF FF FF FC 67 F8 00 7F 80 :D3
7630 AA 00 03 FF FF FF FF 03 FF FF FE 0A A8 00 00 0F :69
7640 FF FF FF F0 FF FF F8 2A A0 00 5B 60 FF FF FE 20 :84
7650 00 00 02 AA 00 00 DB 6D FF FF FC 60 2A AA AA AE :7A
7660 00 02 DB 6D FF FF C1 60 03 FF AB F8 00 06 DB 6D :5C
7670 FF FC 1B 68 00 01 FF 80 00 36 DB 6D FF C0 DB 6C :82
7680 00 00 00 00 01 B6 DB 6D FC 16 DB 6D 80 00 00 00 :D9
7690 05 B6 DB 6D E1 B6 DB 6D B0 00 00 00 1D B6 DB 6D :AD
76A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
76B0 F0 08 08 1F FF FF FF FF FF FF FF F0 1A AA AA 88 :FE
76C0 7F FF FF FF FF FF FC 8A DA AA AA AD 1F FF FF FF :F7
76D0 FF FF C0 AA DA 28 0A A9 43 FF FF FF FF FF 0A 8A :EF
76E0 00 00 00 89 01 FF FF FF FF F8 A8 00 00 00 00 00 :26
76F0 00 7F FF FF FF E2 80 00 00 00 A8 00 00 3F FF FF :C3
---------------------------------------------------------
Sum: 81 74 75 CC 3F 6F 30 13 6F F5 A7 96 BB 29 58 9F :A3

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7700 FF CA 00 2A 80 0A A8 00 02 9F FF FF FF AA AA AA :C1
7710 00 08 87 02 8A 87 FF FF FF 2A AA 80 00 03 FF 82 :77
7720 AA A7 FF FF FF 28 80 00 00 FF FF C2 AA A3 FF FF :01
7730 FE 00 00 07 FF FF FF F0 A0 03 FF FF FC 80 01 FF :0F
7740 FF FF FF F8 00 03 FF FF FC 80 7F FF FF FF FF FE :EB
7750 00 01 FF FF FC A3 FF FF FF FF FF FF 00 01 FF FF :97
7760 FC A2 FF FF FF FF FF FF 00 01 FF FF FC A6 DB FF :13
7770 FB FF FF FF C0 19 FF FF FC 86 DB FF F9 FC 00 0F :2F
7780 C0 F9 FF FF FE 06 DA 00 7C 00 01 FF C1 D9 FF FF :A9
7790 FF 07 80 00 3E 00 0E 3F C1 99 FF FF FF 03 18 00 :83
77A0 1E 02 3F FF C1 19 FF FF FE 03 F0 18 1E 0F FF FF :6A
77B0 C3 19 FF FF FC 40 FF FD 0F 1F FF FF C2 39 FF FF :37
77C0 FD 20 EF FC 4F 8F FF FF 87 29 FF FF FE 90 ED F8 :05
77D0 CF EF 1F FF 8F C3 FF FF FF 52 ED F9 8F FF C3 FF :B3
77E0 0E 1B FF FF FF 88 ED F3 0F FF F8 FF 1E FF FF FF :AE
77F0 FF C8 6F E6 1F E3 FF FF 3E 7F FF FF FF E4 3F FE :F7
---------------------------------------------------------
Sum: 16 27 BB 04 B8 92 F2 16 B5 85 D1 47 E3 08 85 26 :36

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7800 20 1F C7 FF FE 3F FF FF FF F0 1F FE AB BB F8 FF :A9
7810 FE 1F FF FF FF FF 1F FF 6E ED BF 1F FE 8F FF FF :FB
7820 FF FF CF F9 ED DB BF FF FC 87 FF FF FF FF EF F3 :AD
7830 B7 7B FF FF FD C1 FF FF FF FF FB E7 E0 00 03 F3 :A2
7840 F9 C0 1F FF FF FF F8 F3 CF FF F1 E1 F3 C0 00 FF :12
7850 FF FF FE 39 FE 00 FF F3 E7 C0 0A 00 FF FF FF 8F :62
7860 FF FF FF E7 0F 80 1B 6D FF FF FF C7 3F FF FF CC :C8
7870 7E 00 1B 6D FF FF FF E1 9F FF FF 83 F8 00 5B 6D :C4
7880 FF FF FF F0 4F FF FF 0D F0 00 DB 6D FF FF FF F4 :70
7890 13 FF FE 55 C0 00 DB 6D FF FF FF FC 41 FF F0 5E :F4
78A0 00 02 DB 6D FF FF FF FC 10 00 05 70 00 06 DB 6D :16
78B0 FF FF FF F0 35 55 55 80 00 16 DB 6D FF FF FF C0 :67
78C0 03 F5 58 00 00 36 DB 6D FF FF FF 00 00 00 00 00 :CB
78D0 01 B6 DB 6D FF FF F8 30 00 00 00 00 05 B6 DB 6D :28
78E0 FF FF C1 B6 00 00 00 00 0D B6 DB 6D FF FC 19 B6 :4A
78F0 D0 00 00 00 6D B6 DB 6D FF FF FF FF FF FF FF FF :33
---------------------------------------------------------
Sum: 2D 1F 96 47 A1 96 C9 30 C6 E9 64 E0 F3 BB FE 4C :44

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7900 FF FF FF FF FF FF FF FF 80 00 00 00 7F FF FF FF :F4
7910 FF FF FF F0 00 00 00 00 01 FF FF FF FF FF FF 80 :68
7920 00 00 00 00 00 7F FF FF FF FF FC 00 00 00 00 00 :77
7930 10 1F FF FF FF FF E0 00 00 00 00 00 32 0F FF FF :4A
7940 FF FF 00 00 00 00 00 00 42 07 FF FF FF FC 00 00 :40
7950 00 00 00 00 88 01 FF FF FF F8 00 00 00 00 00 04 :82
7960 20 04 FF FF FF F0 00 00 00 00 00 00 40 14 7F FF :E3
7970 FF E0 00 00 00 00 00 00 00 15 3F FF FF C0 00 00 :F1
7980 00 C0 00 00 04 15 1F FF FF 80 00 00 7F 02 00 00 :F7
7990 2C 11 0F FF FF 00 00 1F F8 04 03 03 6C 00 07 FF :DD
79A0 FF 00 01 FF F8 08 0E 1B 6D 00 07 FF FF 00 01 EC :87
79B0 07 FC 7F F0 6D 80 03 FF FF 80 01 C0 00 7F FF 00 :1F
79C0 0D 80 03 FF FF 80 01 88 FC 07 FC 0C 01 80 01 FF :23
79D0 FF 80 01 FF FF C3 E8 7F E1 A0 01 FF FF 80 07 E0 :8F
79E0 00 33 A1 80 03 A0 07 FF FF 90 07 E3 81 3B CE 30 :30
79F0 3F A0 0F FF FF 9C 07 FF C1 FB DC F8 7F B0 2F FF :7B
---------------------------------------------------------
Sum: A9 A0 3F 58 ED 8A 04 3A C1 48 24 A5 D8 49 88 7A :8A

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7A00 FF 9E 0F FF FF E7 DF FF F7 B0 2F FF FF 99 0F FF :EA
7A10 FF E7 9F FF FF B1 2F FF FF 88 8F FF FF E7 97 FF :F3
7A20 FF 21 4F FF FF CE 4F FF FF E7 9F FF FC 65 1F FF :8C
7A30 FF F7 E3 FF FF CF AD FF FC C5 3F FF FF F8 E1 FF :28
7A40 FF 1F B9 FF F1 8C 7F FF FF FE 36 7F FF BF EC FF :2C
7A50 C5 80 FF FF FF FF 85 1F FF F1 9F FD 95 8F FF FF :93
7A60 FF FF E1 87 FF FF FF F5 95 0F FF FF FF FF E1 B3 :8C
7A70 FF FB DF F5 95 0F FF FF FF FF F2 B9 FF FF FF F7 :0D
7A80 55 1F FF FF FF FF F0 B7 FC 00 00 0D 54 3F FF FF :B1
7A90 FF FF FC B6 F1 BF FF 89 50 7F FF FF FF FF FC 36 :E5
7AA0 FF 80 00 F5 50 FF FF FF FF FF FF 16 FF FF FF D5 :A6
7AB0 61 FF FF FF FF FF FF 8E DF FF FF D5 41 FF FF FF :D9
7AC0 FF FF FF 00 DF 15 55 7F 03 FF FF FF FF FF FF 94 :56
7AD0 0F 5B 6D 7C 33 FF FF FF FF FF FF 95 50 DB 6D A0 :4D
7AE0 A0 3F FF FF FF FF FE 87 54 00 00 0A 80 00 3F FF :7C
7AF0 FF FF FE 00 F5 55 54 7A 00 16 C0 01 FF FF F0 00 :D9
---------------------------------------------------------
Sum: 1F 6B BB 9A C5 F2 9F 5A 03 72 1D C6 EC 3E 05 E0 :F6

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
```

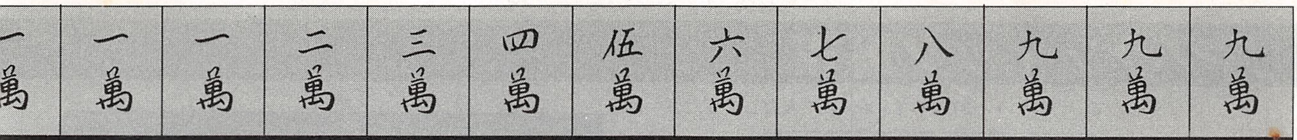
```
7B00 1F F5 57 80 00 36 DB 64 FF FE 00 00 00 3D 50 00 :EA
7B10 00 B6 DB 6D FF F0 74 00 00 00 00 00 01 B6 DB 6D :60
7B20 FC 05 B6 80 00 00 00 00 0D B6 DB 6D 01 6D B6 D0 :36
7B30 00 00 00 00 6D B6 DB 6D DB 6D B6 DB 40 00 00 01 :85
7B40 6D B6 DB 6D DB 6D B6 DB 6D B6 DB 6D 6D B6 DB 6D :1A
7B50 FF FF FF FF FF FF FF FF FF FF FF FE FF FF FF FF :EF
7B60 FF 80 00 01 FF FF FF FE FF FF FF FF 80 00 00 00 :F7
7B70 07 FF FF FE FF FF FF F0 00 00 00 06 52 3F FF FE :84
7B80 FF FF FF 00 00 00 00 06 52 0F FF FE FF FF FB 00 :57
7B90 00 00 00 02 52 43 FF FE FF FF E0 00 00 00 00 00 :72
7BA0 02 40 7F FE FF FF 10 00 00 00 00 00 00 40 7F FE :8A
7BB0 FF F8 10 00 00 00 00 00 00 00 1F FE FF C0 90 00 :73
7BC0 00 00 00 00 00 00 1F FE FF 04 80 00 00 00 00 00 :A0
7BD0 00 00 0F FE FE 04 80 00 00 00 00 00 00 00 1F FE :AC
7BE0 FC 54 00 00 00 00 00 00 00 00 0F FE F8 50 00 00 :A5
7BF0 00 00 01 50 00 00 07 FE F0 D0 10 00 00 00 00 55 :7B
---------------------------------------------------------
Sum: 89 6F 5F 26 93 8C 92 99 92 B7 07 B2 76 A3 E0 F9 :BB

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7C00 00 00 03 FE F0 D0 12 00 00 00 03 40 00 00 01 FE :15
7C10 F0 C0 12 40 00 00 00 00 00 00 01 FE F0 C0 12 40 :03
7C20 00 00 00 00 00 80 01 FE F0 C0 02 00 1F FC 00 00 :4C
7C30 0A 80 01 FE F0 40 01 54 7F FF FF FF AA 80 01 FE :B3
7C40 F0 00 05 55 7F FF FF FF EA A0 03 FE F0 00 05 5F :A5
7C50 FF FF FF FF FE A0 03 FE F0 00 14 00 00 3F FF 80 :5D
7C60 05 A0 03 FE F4 00 14 C8 00 13 F0 00 00 A0 07 FE :1E
7C70 FC 40 15 90 00 01 E0 00 9F A0 0F FE FC 50 15 18 :87
7C80 41 85 E6 23 CE A0 1F FE FC 00 15 FA 07 A5 E6 81 :78
7C90 FE A0 3F FE FC 18 15 AF EF 25 E7 FF FE A1 7F FE :C9
7CA0 FC 31 15 FA AE 25 E5 2B FE A5 7F FE FE 31 15 FF :82
7CB0 F3 25 E5 7F FE CC FF FE FF 05 55 7F E9 25 E0 7F :88
7CC0 FE DC FF FE FF A0 55 7F C9 05 F3 17 FB 99 FF FE :B3
7CD0 FF FC 55 5F 99 F7 E3 87 EB 23 FF FE FF FC 55 5F :63
7CE0 3F 8F 8B F7 EB 1F FF FE FF FE 15 5F 7E B8 3F FF :3C
7CF0 4A 3F FF FE FF EF 15 5D B6 ED BF FF 6C 80 0F FE :40
---------------------------------------------------------
Sum: 9E 40 2F 0A 49 7E 6E 4E 39 F4 B1 22 75 D4 30 88 :9B

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7D00 F7 F8 15 55 A7 ED BF FF 62 DA 03 FE BF E0 05 55 :E1
7D10 A0 00 00 35 86 DB 69 FE FF 80 01 55 9B DF F9 97 :7C
7D20 06 DB 6C 3E F0 00 08 15 7C 00 05 B0 06 DB 6D 8E :A5
7D30 81 00 0E 05 5F 5B 6D B0 06 1B 6D A2 03 00 03 81 :22
7D40 5F FF ED D0 16 C3 6D B6 1B 00 00 D0 1F FF FE 00 :1E
7D50 1A D3 6D B6 C3 40 00 0C 00 00 00 00 30 CB 6D B6 :3D
7D60 D9 60 00 00 00 00 00 00 66 9B 6D B6 DB 6D 00 00 :A5
7D70 00 00 00 05 D6 1B 6D B6 DB 6D 80 00 00 00 00 6D :4E
7D80 B4 DB 6D B6 DB 6D B0 00 00 00 1B 6D B6 DB 6D B6 :E6
7D90 DB 6D B6 D8 00 00 DB 6D B6 DB 6D B6 DB 6D B6 DB :AB
7DA0 6D B6 DB 6D B6 DB 6D B6 FF FF FF FF FF FF FF FF :17
7DB0 FF FF FF FE FF FF FF FF 80 00 00 00 FF FF FF FF :73
7DC0 FF FF FF C0 00 00 00 00 07 FF FF FF FF FF F8 00 :B7
7DD0 00 00 00 00 00 3F FF FF FF FF 00 00 00 00 00 00 :3B
7DE0 00 03 FF FF FF FC 00 00 00 00 00 00 00 00 7F FF :7A
7DF0 FF F8 00 00 00 00 00 00 00 00 3F FF FF E0 00 00 :14
---------------------------------------------------------
Sum: 69 FC E4 10 BA C3 6D 5B 7A 55 28 4B 1A F6 71 AC :0D

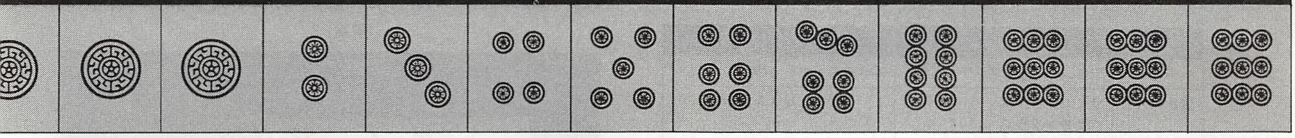
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7E00 00 00 00 00 00 00 0F FF FF C0 00 00 00 00 00 00 :CD
7E10 00 00 01 FF FF 00 00 00 00 00 00 00 00 00 00 FF :FE
7E20 FF 00 00 00 00 00 00 00 00 00 00 1F FD 00 00 00 :1B
7E30 00 00 00 00 00 00 00 0F FC 00 00 00 00 00 00 00 :0B
7E40 00 00 00 07 F8 00 00 00 00 00 00 00 00 00 00 03 :02
7E50 F8 00 00 00 00 00 00 00 00 00 00 03 F8 00 00 00 :F3
7E60 00 00 00 00 00 00 00 03 F8 00 00 00 00 00 00 00 :FB
7E70 00 00 00 01 FC 00 00 00 00 00 00 00 00 00 00 01 :FE
7E80 FC 00 00 00 00 00 00 00 00 00 00 01 FF 80 00 1B :97
7E90 01 B6 C0 00 00 00 00 01 FF 80 00 DB 6D B6 DB 00 :D0
7EA0 00 00 00 01 FF E0 00 DB 6D B6 DB 60 00 00 00 01 :1A
7EB0 FF E0 00 5B 6D B0 01 60 00 00 00 01 FF F0 02 BB :35
7EC0 6D 80 3C 20 00 00 00 01 FF FF 06 C1 6C 03 01 A0 :1F
7ED0 00 00 00 03 FF FF 84 00 6C 00 00 20 00 00 00 03 :14
7EE0 FF FF E5 88 6C 2E 10 84 00 00 00 07 FF FF E0 84 :02
7EF0 70 C8 42 B5 40 00 00 07 FF FF F0 C9 70 DB 6D B5 :9A
---------------------------------------------------------
Sum: CF DD 24 C3 0A BD A4 D9 C9 F4 D1 10 3B 03 2B B6 :64

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7F00 10 00 00 07 FF FF F2 D8 F0 DB 6D B5 10 00 00 0F :EB
7F10 FF FF F2 D8 E0 1B 6D 85 08 00 00 0F FF FF FA E2 :A6
7F20 E0 0B 6D 85 C6 00 00 1F FF FF F9 A1 D6 CB 6D 15 :7D
7F30 46 00 00 3F FF FF F9 E0 06 8B 69 54 87 00 00 7F :B0
7F40 FF FF FE DD B6 D9 21 51 03 80 01 FF FF FF FE 2C :85
7F50 36 B9 05 43 03 00 07 FF FF FF FE C5 FE B1 55 53 :58
7F60 03 00 3D FF FF FF FF 68 40 01 55 13 BA 03 EF FF :F8
7F70 FF FF FF 84 DE D5 55 04 38 FF FF FF FF FF FF FA :B9
7F80 00 15 54 00 42 91 FF FF FF FF FF FE 36 15 40 02 :C2
7F90 40 80 03 FF FF FF FF FF 0E D9 00 16 00 00 D0 7F :0A
7FA0 FF FF FF FF E0 00 00 12 40 02 DB 0F FF FF FF FF :13
7FB0 FF FC 2C 32 00 06 DB 07 FF FF FF FF FF E0 6D 32 :BB
7FC0 00 36 DB 43 FF FF FF FF FE 08 6D 80 01 B6 DB 61 :36
7FD0 FF FF FF FF FC 18 68 00 0D B6 DB 60 FF FF FF FF :72
7FE0 E0 1B 40 00 2D B6 DB 6C FF FF FF FF 80 00 00 00 :E1
7FF0 6D B6 DB 6C FF FF FF FE 00 00 00 03 6D B6 DB 6C :D2
---------------------------------------------------------
Sum: F6 57 15 24 82 28 EE 98 CD 7A 3F 93 43 DB D9 7B :41
```



FM-8 フルカラー版
グラフィック麻雀

●峰岸順二



麻雀の魅力は，テンパイが近くなってきたときのつぎのツーモ牌への期待，テンパイしてから和了するまでの，あの何とも言えないスリルではないでしょうか．

この醍醐味(だいごみ)は，いままで発表されたＢＳ，ＰＣ，Ｌ３，MZ-80Bなどの４人麻雀ゲームで充分満喫できるのですが，ちょっとものたりないようです．やはり，色とりどりのレイアウトが麻雀を一層，楽しくしています．

その名はFM-8，640×200ドットのグラフィック機能，各ドットに色指定可能，これをフルに使ってパイの色とデザインを実戦のものとまったく同じにレイアウトした，FM-8グラフィック麻雀を完成したので発表します．緑一色も可能です．ぜひ挑戦してみてください．

ゲームの特徴

TK-80BS ４人麻雀（I/O 54年12月）のアルゴリズムを使った麻雀ゲームは，何度か本誌に紹介されていますが，バック・ナンバーが必要のないように，ここに，特徴と遊び方を一度整理します．

❶ 上家，対面，下家の３人をマイコンが担当し，この３人を相手に勝負を進めます．４人がお互いに，『振り込み』や『ツモロン』をして，実戦とまったく同じです．もちろん，マイコン内部のＲＡＭにある３人は独立しており，イカサマなどすることは絶対にありません．

❷ 敵３家も『テンパイ』すると適当なときに『リーチ』をかけてきますが，『ヤミテン』もしています．『チー』，『ポン』もできますが，敵３家は面前だけです．

❸ サイフリもマイコンが行ない，あなたの風もマイコンが指示してくれます．

❹ 和了すると和了牌がディスプレイされ，『ドラ』と『ウラドラ』も出力されます．基礎点（得点）と何番かを入力すると，マイコンが計算し，４人の点数を出力します．

❺ 18巡して和了がないときは流局になります．ただ実戦と違う所はカンがありません．プログラムが長くなるので割愛しました．

遊び方

RUNすると，まずピンズ牌のデモがあります．つぎに画面が変わり，『牌をよくかき混ぜるために，５～10秒の後に スペース キーを押してください』と出るので スペース キーを押してください．すると相手の強さをどの程度にしたらよいかを聞いてきます．

最高クラス	プロがお相手します……	1
平均クラス	一般アマチュア程度……	2
入門クラス	初心者がお相手…………	3

1，2，3 のいずれかを入力してください．

洗牌(パイのかき混ぜ)，城壁牌（チョンピーパイ，牌を井ゲタ型に並べる），および骰子（シャイツ）がふられて配牌するのに約30秒かかるので待ってください．そして画面一転，いよいよゲーム開始です．局数，門風(あなたの風)，ドラパイの表示の後，美しい緑の卓子（トオズ，テーブル）の上にあなたの配牌がきちんと理牌されてディスプレイされます．他家は順々に自模（ツモ)，もちろんこれは見えません．打牌（ターパイ，捨て牌）だけが出力されます．

手牌（ショーパイ，自分の牌）の下にa～nが出るので，捨て牌やチー，ポンのときはこのアルファベットを入力し RET してください．

他３家は自動的にツーモ，ツモロンのチェック，リーチの宣言，捨て牌，他家へのフリコミ，流れのチェックを一瞬で行ない，下家へバトン・タッチします．

下家，対面，上家とスピーディにつぎつぎとパイが打たれ，この間，ツモロンやフリコミがあれば，『何家のフリコミ』，『何家のロン』と表示され，和了した家のパイがディスプレイされます．

他家の捨て牌でチー，ポン，またはロンするときはまず スペース キーを押すと『ショチ？』と聞いてきます．

『チイ』『ポン』のときは C または P を入力し，あなたの食う牌を聞いてくるので，１枚ずつa～nのアルファベットを入力します．つぎに打牌するのは実戦と同じで，ハダカタンキになるまでチイ，ポンできます．

ロンのときは R を入力すれば，和了牌とドラ，ウラドラ牌が出力されます．基礎点(あがり点＋副底20符，得点）および何翻かを入力すればマイコンが計算します．

面前でテンパイしてリーチの場合は『**リーチ**』と入力し，つぎに捨て牌をａ～ｎとして入力します．待ち牌は２枚まで入力できます．『**マチハイー１？**』『**マチハイー２？**』と聞

いてくるので，つぎの１～37のパイ・コードで入力してください．

１ピン―９ピン	１―９
１万―９万	11―19
１索―９索	21―29
東南西北	31―34
緑発，紅中，白板	35―37

リーチをかければ，マイコンがすべてロンのチェックをしてくれます．

遊び方をまとめて図１に示します．文を読むと面倒のようですが，やってみると意外と簡単です．27,000点配点の30,000点返し，満貫は親12,000，子8,000です．東の４局が終るとまた東の１局になります．

図１　遊び方

アクション	ディスプレイの出力	入　力
RUN	Space Keyヲオシテクダサイ	スペース・キーを押す
強さの入力	サイコウクラス　1 ヘイキンクラス　2 ニュウモンクラス　3	１，２，または３を入力
捨て牌	ステハイ？	a～nのいずれかを入力
チー，ポン	 ショチ？ クイハイー１？ クイハイー２？ ステハイ？	スペース・キーを押す チー＝C，ポン＝Pを入力 チー，ポンの第１牌a～nを入力 同上第２牌a～nを入力 a～nのいずれかを入力
ロン	ショチ？	Space Keyを押す ロン＝rを押す
リーチ	ステハイ？ マチー１？ マチー２？	「リーチ」を入力 a～nのいずれかを入力 待ち牌パイコードで入力 ピンズ　1～9 マンズ　11～19 ソーズ　21～29 東南西北　31,32,33,34 発中白　35, 36, 37 同上
点数計算	キソノテンスウ ナンハンデスカ	基礎符の入力 翻数の入力

グラフィック麻雀の改良

Ｉ/Ｏ，'81年12月号に発表された『グラフィック麻雀』の改良版が送られてきました．以下，改良された点とプログラム・リストを掲載します．

編集部

写真１



写真２

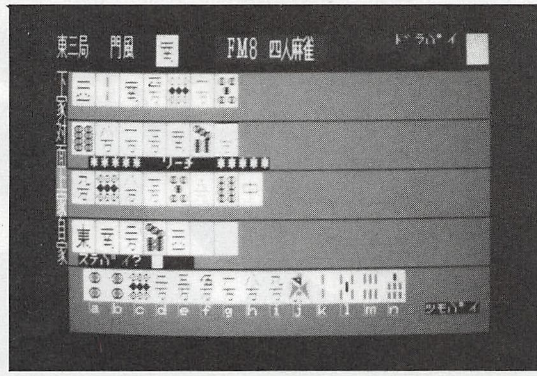


写真３

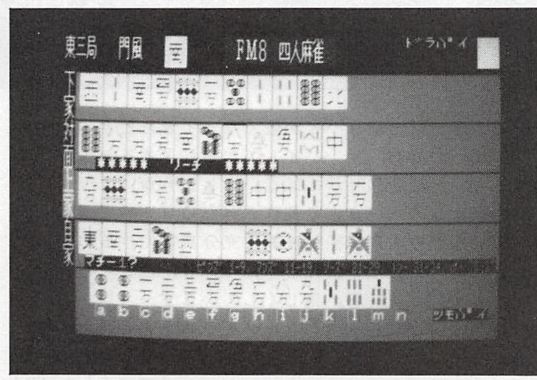
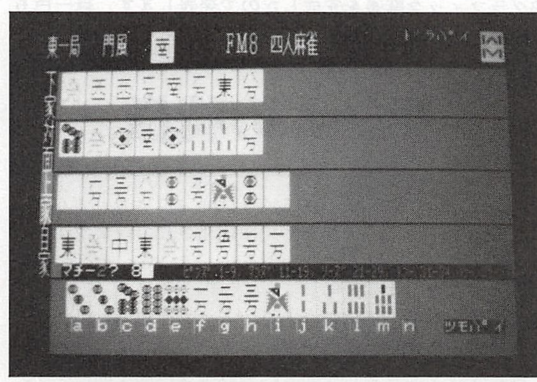


写真４



Ｉ/Ｏに発表された峰岸氏のプログラムをキーボード入力してから日を置いてぽつりぽつりとプログラムの中味をいじってきました．いろいろな視点から見直しては多少思い切った改造もやってみました．その結果，もとのプログラムとの対応がはっきりとはつけられなくなってしまいました．そこで改造については，❶どのような方針で改造したか．❷操作上変わった所はどこか．❸表示上変わった所はどこかを示します．

処理速度

ゲーム・プログラムは処理速度を速くして対戦者の気がそれることを防がなくてはなりません．その第一はプログラムのアルゴリズムを充分に解析して，同一処理結果を得るのであればより処理時間の短いプログラムに改造するこ

と．第二は対戦者が画面から情報を読み取るなど対戦者のつごうで画面を固定している時間を利用して裏で適当な量の処理をやってしまうこと．

第一の方法により配牌や点数計算プログラムの改造をしました．

第二の方法により，メモリ・クリアやE(L)切りプログラムを二つの画面の間にはさんで心理的に気にならなくしました．

キー入力のうち，ワン・キー入力で済むものは極力リターン・キーなしで行なえるようにしました．FM-8が入力待ちであることを示すため，黄色いカーソル・ポイントが点滅するようにしてあります．これはインプット入力なので，早目のキー入力もある程度は受け付けます．たとえばRUN直後に[1]，[2]または[3]の数字をキー入力すると，表紙デモ，ゲームの説明の後すぐに入力強さ処理を済ませて次の処理に進みます．

他家のパイ出力のあと[PF.1]をキー入力すると直ちに2～4秒の時間待ちを中止して次の家に処理を進めます．

表示処理

画面表示上でうるさく見える部分をなくし，必要な部分を追加しました．

- デモ表示のマンズの表示がもたついているのはサークル命令を多用しているためなので，**GET, PUT**命令を併用することにより処理速度を速めました．ソーズの表示も一部改造しました．
- 自家の牌をツモ・キリなどのたびに全面的に書き直しているのは時間の無駄が多いので，必要最少限の書き直しにとどめます．
- リーチの際には捨て牌を赤枠で囲んではっきりさせます．自家リーチの際には待ち牌コードを調べるために，画面に牌コード表を出します．
  END表示がなかったので追加しました．
- チー，ポンされたパイを完全に消すと食いハイの選択に不便なので，パイの色を目立たない色に変更するにとどめました．
- 親になった家の名を紫色にしました．
- チー，ポンした後の捨てパイがそれ以前の捨てパイの上に重なる場合があったので，完全に重ならないようにしました(これは変色パイ保存のため)．
- 自家(人間)がロンした場合はせっかく並んでいるパイを消して再出力するのはもったいないので**GET, PUT**文であがりパイの表示へ移動することにしました．
- 点数計算はあがりパイの表示画面を残したままで行なうようにしました．

操作性

キー入力の心要性を少しでも少なくして，入力ミスの救済措置をします．

- RUN直後の[SPACE]キー入力は省略し，乱数初期値設定はTIMEを使いました．
- 捨て牌，食い牌を指定するためのa～nの入力ミスを防ぐために，大文字・小文字いずれの入力も可能としました．また，a～nのうちチー・ポンなどで欠落した部分，もしくは全然関係ないキーを入力した場合にはビープ音を鳴らした後に再入力待ちになるようにしました．
  これは通常の捨て牌，チー・ポンの食い牌のいずれにも適用できるようにしました．リーチの待ち牌コードのミスは救済していません．

その他

単純変数はすべてRUN直後に定義して，その後に配列変数を定義しました．このようにすると実行速度がわずかですが速くなります．

- 変数を整理して少なくしました．実行速度がわずかに速くなります．
- マルチ文を用いました．実行速度がわずかに速くなります．
- サブルーチンを減らしました．実行速度がわずかに速くなります．
- **FOR～NEXT**文を追加したり，減らしたりしました．実行速度がわずかに速くなります．
- プログラムの多機能化によりメモリの残量が少なくなってしまいました．
  カセットテープ・ベースでは問題ありませんが，フロッピーディスク・ベースで動くシステムではメモリ・オーバーフローのおそれが出てきたので，プログラムの刈り込みをしました．その方法は，㋐他家のロン・チェックでロン成立の判断からロン処理プログラムへのジャンプを**RETURN×××**という形で行ない，多所か**IF Q1 THEN×××**を消滅させました．㋑"カミチャ"，"シチチャ"などをS＄(4)の配列文字変数に入れることで，フリコミとロンの家の表示から**IF**文を消滅させました．同様にH3を2～12だったのを1～4にすることで**ON GO TO**文などを縮小しました．㋒サブルーチンを統廃合しました．
- 変数名をいくつか統廃合しました．
- プログラムの内容がかなり変わったので，ラインナンバーの整理を行ないました(**RENUN**による)．
- 点数計算のロジック・アルゴリズムはもっと複雑なはずですが，よく解らないので，そのままにしておきました．なお，変数N3はゲーム続行中にゼロ復帰しないようにしておいたので何か役に立つかも知れません．

グラフィック麻雀プログラム・リスト

```
0 GOTO4100:'** FM-8 ｸﾞﾗﾌｨｯｸ 4ﾏｰｼﾞｬﾝ *** 57年 5月 17日 ﾔﾅｷﾞﾉ ﾏｻﾌﾐ ***
         *** 56年 10月 12日 ﾐﾈｷﾞｼ ｼﾞｭﾝｼﾞ *** ｸﾗｼｷﾏｲｺﾝｸﾗﾌﾞ ｷﾏｺﾝｸﾗﾌﾞ FORESIGHT
1 PUT@A(X+1,Y+4)-(X+31,Y+16),Z1,PSET:RETURN
2 PUT@A(X+4,Y+2)-(X+28,Y+18),Z2,PSET:RETURN
3 PUT@A(X,Y)-(X+31,Y+18),Z3,PSET:RETURN
4 PUT@A(X,Y+3)-(X+31,Y+17),Z4,PSET:RETURN
5 PUT@A(X,Y)-(X+31,Y+18),Z5,PSET:RETURN
6 PUT@A(X,Y)-(X+31,Y+18),Z6,PSET:RETURN
7 PUT@A(X,Y)-(X+31,Y+18),Z7,PSET:RETURN
```

プログラム・リスト

```
8 PUT@A(X,Y)-(X+31,Y+18),Z8,PSET:RETURN
9 PUT@A(X,Y)-(X+32,Y+18),Z9,PSET:RETURN
11 CONNECT(X+6,Y+4)-(X+25,Y+4),0:GOTO20
12 CONNECT(X+9,Y+3)-(X+22,Y+3),0:CONNECT(X+6,Y+7)-(X+25,Y+7),0:GOTO20
13 CONNECT(X+8,Y+2)-(X+22,Y+2),0:CONNECT(X+10,Y+5)-(X+20,Y+5),0:CONNECT(X+6,Y+8)
-(X+25,Y+8),0:GOTO20
14 LINE@(X+6,Y+2)-(X+25,Y+7),PSET,0,B:CONNECT(X+12,Y+3)-(X+12,Y+5)-(X+6,Y+6),0:C
ONNECT(X+18,Y+3)-(X+18,Y+5)-(X+24,Y+5),0:GOTO20
15 CONNECT(X+13,Y+2)-(X+6,Y+6),0:CONNECT(X+10,Y+4)-(X+10,Y+8),0:CONNECT(X+14,Y+2
)-(X+23,Y+2),0:CONNECT(X+19,Y+3)-(X+16,Y+8),0:CONNECT(X+14,Y+5)-(X+24,Y+5)-(X+22
,Y+8),0:CONNECT(X+12,Y+8)-(X+25,Y+8),0:GOTO20
16 CONNECT(X+15,Y+2)-(X+15,Y+3),0:CONNECT(X+7,Y+4)-(X+24,Y+4),0:CONNECT(X+11,Y+6
)-(X+8,Y+8),0:CONNECT(X+20,Y+6)-(X+22,Y+8),0:GOTO20
17 CONNECT(X+6,Y+4)-(X+25,Y+4),0:CONNECT(X+13,Y+2)-(X+13,Y+8)-(X+22,Y+8)-(X+22,Y
+7),0:GOTO20
18 CONNECT(X+12,Y+3)-(X+6,Y+8),0:CONNECT(X+17,Y+2)-(X+19,Y+2)-(X+20,Y+5)-(X+25,Y
+8),0:GOTO20
19 CONNECT(X+15,Y+1)-(X+15,Y+4)-(X+13,Y+6)-(X+9,Y+8),0:CONNECT(X+9,Y+3)-(X+21,Y+
3)-(X+19,Y+7)-(X+20,Y+8)-(X+26,Y+8)-(X+26,Y+7),0
20 CONNECT(X+6,Y+11)-(X+25,Y+11),2:CONNECT(X+14,Y+11)-(X+11,Y+16)-(X+8,Y+17),2:C
ONNECT(X+13,Y+14)-(X+22,Y+14)-(X+20,Y+18)-(X+15,Y+18),2:RETURN
21 IF Z21(1) THEN PUT@A(X,Y)-(X+32,Y+19),Z21,PSET:RETURN ELSE100
22 X1=X+14:LINE@(X1,Y+2)-(X1+4,Y+8),PSET,4,BF:LINE@(X1,Y+11)-(X1+4,Y+17),PSET,4,
BF:RETURN
23 CL=4:X1=X+14:Y1=Y+2:X2=4:Y2=6:GOSUB30:X1=X+7:Y1=Y+12:GOSUB30:X1=X+21:GOTO30
24 CL=4:X2=4:Y2=6:X1=X+7:Y1=Y+2:GOSUB30:Y1=Y+12:GOSUB30:X1=X+21:GOSUB30:Y1=Y+2:G
OTO30
25 CL=4:X2=4:Y2=6:X1=X+6:Y1=Y+2:GOSUB30:Y1=Y+12:GOSUB30:X1=X+22:GOSUB30:Y1=Y+2:G
OSUB30:CL=2:X1=X+14:Y1=Y+7:GOTO30
26 CL=4:X2=4:Y1=Y+2:Y2=16:FOR X1=X+6 TO X+22 STEP 8:GOSUB 30:NEXT:LINE@(X+6,Y+9)
-(X+26,Y+11),PSET,7,BF:RETURN
27 CL=4:X2=4:Y2=10:Y1=Y+8:FOR X1=X+7 TO X+23 STEP 8:GOSUB30:NEXT:LINE@(X+7,Y+13)
-(X+27,Y+13),PSET,7:CL=2:X1=X+15:Y1=Y+2:Y2=4:GOTO30
28 CL=4:X2=3:Y2=5:X1=X+3:Y1=Y+12:GOSUB30:Y1=Y+3:GOSUB30:X1=X+26:GOSUB30:Y1=Y+12:
GOSUB30:CONNECT(X+8,Y+7)-(X+16,Y+4)-(X+23,Y+8)-(X+24,Y+7)-(X+17,Y+4)-(X+9,Y+8),4
:CONNECT(X+8,Y1)-(X+16,Y+16)-(X+24,Y1)-(X+24,Y1+1)-(X+17,Y+16)-(X+9,Y1),4:RETURN
29 IF Z29(1) THEN PUT@A(X+4,Y+2)-(X+28,Y+18),Z29,PSET:RETURN ELSE120
30 LINE@(X1,Y1)-(X1+X2,Y1+Y2),PSET,CL,BF:RETURN
31 CONNECT(X+6,Y+6)-(X+26,Y+6),0:CONNECT(X+15,Y+2)-(X+15,Y+17)-(X+16,Y+17)-(X+16
,Y+2),0:LINE@(X+8,Y+8)-(X+24,Y+12),PSET,0,B:CONNECT(X+9,Y+10)-(X+24,Y+10),0:CONN
ECT(X+5,Y+17)-(X+11,Y+16)-(X+15,Y+11)-(X+21,Y+16)-(X+26,Y+17),0:RETURN
32 CONNECT(X+6,Y+5)-(X+25,Y+5),0:CONNECT(X+15,Y+3)-(X+15,Y+8),0:CONNECT(X+6,Y+17
)-(X+6,Y+8)-(X+25,Y+8)-(X+25,Y+17)-(X+22,Y+17),0:CONNECT(X+11,Y+8)-(X+15,Y+13)-(
X+21,Y+8),0:CONNECT(X+10,Y+12)-(X+22,Y+12),0:GOTO130
33 CONNECT(X+6,Y+5)-(X+25,Y+5),0:LINE@(X+6,Y+9)-(X+25,Y+17),PSET,0,B:CONNECT(X+1
3,Y+5)-(X+13,Y+13)-(X+6,Y+14),0:CONNECT(X+18,Y+5)-(X+18,Y+13)-(X+25,Y+14),0:RETU
RN
34 CONNECT(X+12,Y+4)-(X+12,Y+15),0:CONNECT(X+6,Y+10)-(X+10,Y+10),0:CONNECT(X+4,Y
+17)-(X+12,Y+16),0:CONNECT(X+19,Y+4)-(X+19,Y+17)-(X+27,Y+17)-(X+27,Y+15),0:CONNE
CT(X+19,Y+11)-(X+26,Y+9),0:RETURN
35 CONNECT(X+11,Y+4)-(X+18,Y+4)-(X+4,Y+12),4:CONNECT(X+8,Y+6)-(X+12,Y+8),4:CONNE
CT(X+15,Y+6)-(X+25,Y+12)-(X+28,Y+12),4:CONNECT(X+22,Y+6)-(X+19,Y+8),4:CONNECT(X+
24,Y+8)-(X+21,Y+10),4:GOTO140
36 LINE@(X+6,Y+7)-(X+25,Y+12),PSET,2,B:CONNECT(X+5,Y+7)-(X+5,Y+13),2:CONNECT(X+2
6,Y+7)-(X+26,Y+13),2:CONNECT(X+15,Y+3)-(X+15,Y+18)-(X+16,Y+18)-(X+16,Y+3),2:RETU
RN
37 RETURN
100 X1=X+16:Y1=Y+10:X2=X+1:FOR Y2=Y+6 TO Y+18 STEP 2:CONNECT(X1,Y1)-(X2,Y2),4:CO
NNECT(X1,Y1)-(X+31,Y2),4:NEXT Y2:X1=X+4:Y1=Y+4:X2=X+19:FOR Y2=Y+3 TO Y+8:CONNECT
(X1,Y1)-(X2,Y2),2:NEXT Y2
110 CONNECT(X+12,Y+14)-(X+12,Y+19)-(X+8,Y+19),0,AND:CONNECT(X+19,Y+14)-(X+19,Y+1
9)-(X+17,Y+19),0:CIRCLE@(X+16,Y+14),5,4,,,,F:PSET(X+14,Y+5,7):PSET(X+15,Y+5,7):L
INE@(X+12,Y+1)-(X+20,Y+3),PSET,3,BF:GET@A(X,Y)-(X+32,Y+19),Z21,G:RETURN
120 X2=4:Y1=Y+2:Y2=16:CL=2:X1=X+15:GOSUB30:CL=4:X1=X+23:GOSUB30:X1=X+7:GOSUB30:L
INE@(X,Y+7)-(X+32,Y+13),PSET,7,B:GET@A(X+4,Y+2)-(X+28,Y+18),Z29,G:RETURN
130 CONNECT(X+10,Y+15)-(X+22,Y+15),0:CONNECT(X+15,Y+12)-(X+15,Y+17),0:RETURN
140 CONNECT(X+8,Y+11)-(X+21,Y+11),4:CONNECT(X+7,Y+14)-(X+22,Y+14),4:CONNECT(X+12
,Y+11)-(X+12,Y+15)-(X+10,Y+17)-(X+4,Y+18),4:CONNECT(X+17,Y+11)-(X+17,Y+18)-(X+24
,Y+18)-(X+24,Y+16),4:RETURN
150 'ﾊﾟｲ ﾉ CRT ｼｭﾂﾘｮｸ
160 LINE@(X,Y)-(X+32,Y+20),PSET,7,BF
170 ON E GOTO 1,2,3,4,5,6,7,8,9,0,11,12,13,14,15,16,17,18,19,0,21,22,23,24,25,26
,27,28,29,0,31,32,33,34,35,36,37
180 '
190 X=INT(L/4)*36+24:Y=J1*32-6:L=L+1:E=E(L):IF P(J1)=0 THEN 250
200 IF L<Q(J1) THEN 250
210 IF Q(J1)=0 THEN 230
220 Q(J1)=0:LINE@(X-1,Y-1)-(X+33,Y+21),PSET,2,B:LOCATE 3,2+J1*4:PRINT"***** ﾘｰﾁ
 *****";
230 BEEP 1:FOR I=1 TO 100:NEXT:BEEP 0:GOTO 260
```

```
240 'ﾛﾝ ﾅｶﾞﾚ ﾁｪｯｸ
250 IF L<RND(1)*65+8 THEN 280
260 IF (E=C(J1,15))+(E=C(J1,16)) THEN490
270 'ｽﾃﾊﾟｲ ﾉ ﾃﾞｨｽﾌﾟﾚｲ
280 GOSUB160:IF P6=4 THEN 350
290 TIME$=TIME$:G=TIME+N7
300 F$=INKEY$:IF F$=" " THEN 320
310 IF TIME>G THEN 350 ELSE 300
320 LOCATE 2,18:PRINT"ｼｮﾁ";:GOSUB2170
330 IF (F$="c")+(F$="C")+(F$="p")+(F$="P") THEN 1990
340 LOCATE2,18:PRINT SPC(5);:IF (F$="r")+(F$="R") THEN J3=4:GOTO500
350 IF L>67 THEN 630
360 IF J1=1 THEN 650
370 IF J1=2 THEN 690
380 GOTO 730
390 'ﾀﾁｬ ﾉ ﾛﾝ ﾁｪｯｸ
400 H=15*N7-5:IF L<RND(1)*(73-H)+H THEN RETURN:'▬ﾂﾖｻ
410 IF (E=C(J3,15))+(E=C(J3,16)) THEN RETURN500
420 RETURN
430 'ﾊﾟｲ ﾉ ｼｭﾂﾘｮｸ
440 X=I1*34+6:X1=482-N5*34:LINE@(X,154)-(X+32,174),PRESET,0,BF:GET@A(X+34,154)-(
X1+34,174),ZA,G:GOSUB2150:PUT@A(X,154)-(X1,174),ZA,PSET:RETURN
450 'ﾂｰﾓ ﾊﾟｲ ｦ ｹｽ
460 LINE@(568,154)-(600,174),PSET,4,BF:RETURN
470 'ｱｶﾞﾘ ﾉ ﾅｶﾞﾚ ｼｮﾘ
480 X=482-N5*34:GET@A(40,154)-(X,174),ZA,G
490 GOSUB4380:CLS:PRINT S$(J1);" ﾉ ﾂﾓ ﾛﾝ";:P1=J1:GOTO 540
500 IF J3=4 THEN X=482-N5*34:GET@A(40,154)-(X,174),ZA,G
510 GOSUB4380:CLS:PRINT S$(J1);" ﾉ ﾌﾘｺﾐ":P3=J1
520 PRINT S$(J3);" ﾉ ﾛﾝ";:J1=J3:P2=J3
530 'ｱｶﾞﾘ ﾉ ﾊﾟｲ ﾉ ﾃﾞｨｽﾌﾟﾚｰ
540 BEEP 1:FOR I=1 TO 400:NEXT:BEEP 0:LINE@(0,20)-(639,76),PSET,4,BF
550 Y=28:IF J1=4 THEN PUT@A(40,Y)-(X,Y+20),ZA,PSET:GOTO570
560 N5=0:X=40:B=E:FOR I1=1 TO 13:E=C(J1,I1):GOSUB160:X=X+34:NEXT:E=B
570 IF P1-4 THEN X=560
580 GOSUB160:LOCATE 2,11:PRINT"ﾄﾞﾗﾊﾟｲ";:E=E(82):X=138:Y=83:GOSUB160:LOCATE 15,11
:PRINT"ｳﾗﾄﾞﾗ";:E=E(83):X=330:GOSUB160:IF N5 THEN Y=48:GOSUB2130
590 LOCATE 9,14:PRINT"ﾃﾝｽｳｹｲｻﾝ=0: ｻｲﾌﾘ=1";:GOSUB2170
600 IF F$="0" THEN 1840
610 IF F$="1" THEN 1100
620 GOTO 590
630 CLS:PRINT"ﾅｶﾞﾚﾃﾞｽ";:FOR I=0 TO 999:NEXT:GOTO1100
640 J1=1:GOTO 190:'▬ｼﾓﾁｬ
650 J3=2:GOSUB400
660 J3=3:GOSUB400
670 J3=4:GOSUB410
680 J1=2:GOTO 190:'▬ﾄｲﾒﾝ
690 J3=3:GOSUB400
700 J3=4:GOSUB410
710 J3=1:GOSUB400
720 J1=3:GOTO 190:'▬ｶﾐﾁｬ
730 J3=4:GOSUB410
740 J3=1:GOSUB400
750 J3=2:GOSUB400
760 L=L+1:J1=4:E=E(L):X=568:Y=154:GOSUB160:GET@A(568,154)-(600,174),ZB,G:'ﾂﾓﾊｲ ﾉ
 ﾃﾞｨｽﾌﾟﾚｰ
770 IF P6-4 THEN 820
780 'ｼﾞﾌﾞﾝ ﾉ ﾘｰﾁ ﾉ ﾛﾝ ﾉ ﾁｪｯｸ
790 IF (E=C(4,15))+(E=C(4,16)) THEN480
800 'ﾘｰﾁ ﾉ ｽﾃﾊﾟｲ ﾉ ﾃﾞｨｽﾌﾟﾚｰ
810 GOSUB460:X=INT((L-1)/4)*36+24:PUT@A(X,122)-(X+32,142),ZB,PSET:GOTO1000
820 FOR I1=13-N5 TO 1 STEP -1
830 C(4,I1+1)=C(4,I1):IF E>=C(4,I1) THEN C(4,I1+1)=E:I1=I1+1:GOTO 850
840 NEXT:I1=1:C(4,1)=E
850 GOSUB460:X=I1*34+6:X1=482-N5*34:GET@A(X,154)-(X1,174),ZA,G:GOSUB2150:PUT@A(X
,154)-(X+32,174),ZB,PSET:PUT@A(X+34,154)-(X1+34,174),ZA,PSET:GOTO870
860 BEEP 1:FOR I=1 TO 400:NEXT:BEEP 0
870 LOCATE 2,18:INPUT"ｽﾃﾊﾟｲ";F$:IF (F$="r")+(F$="R") THEN E=C(4,14-N5):GOTO480
880 IF F$="ﾘｰﾁ" THEN P6=4:GOTO870
890 FOR I1=1 TO 28:IF F$=R$(I1) THEN 910
900 NEXT:GOTO860
910 IF I1>14 THEN I1=I1-14
920 IF I1>14-N5 THEN 860
930 LOCATE2,18:PRINT SPC(10):E=C(4,I1):X=I1*34+6:GET@A(X,154)-(X+32,174),ZB,G:LI
NE@(X,154)-(X+32,174),PRESET,0,BF:B=INT((L-1)/4)*36+24:PUT@A(B,122)-(B+32,142),Z
B,PSET:FOR J=I1 TO 13:C(4,J)=C(4,J+1):NEXT:'ｽﾃﾊﾟｲ ｦ ﾂﾒﾙ
940 GOSUB440
950 IF P6-4 THEN 1010
960 'ﾘｰﾁ ﾉ ﾏﾁ ﾉ INPUT ﾄ ﾘｰﾁ ｾﾝｹﾞﾝ
970 LINE@(200,144)-(634,151),PRESET,0,BF:SYMBOL(202,145),"ﾋﾟﾝｽﾞ 1-9, ﾏﾝｽﾞ 11-19,
```

プログラム・リスト

```
 ｿｰｽﾞ 21-29, ﾄﾝｰ 31-34,ﾊﾂｰ 35-37",1,1,4
980 LOCATE 2,18:INPUT"ﾏﾁｰ1";C(4,15):LOCATE 2,18:INPUT"ﾏﾁｰ2";C(4,16):LINE@(17,144
)-(634,151),PSET,4,BF:LOCATE 3,18:PRINT"*****  ﾘｰﾁ  *****";
990 LINE@(B-1,121)-(B+33,143),PSET,2,B
1000 BEEP 1:FOR I=1 TO 100:NEXT:BEEP 0
1010 H=15*N7-5:IF L<RND(1)*(73-H)+H THEN1050:'■ﾂﾖｻ
1020 J3=1:GOSUB410
1030 J3=2:GOSUB410
1040 J3=3:GOSUB410
1050 IF L>67 THEN630
1060 J1=1:GOTO190
1070 H=INT(RND(1)*37)+1:IF D(H)>=P THEN1070
1080 D(H)=D(H)+5-P:RETURN
1090 'ﾒﾓﾘｰ ｸﾘｬｰ
1100 P1=0:P2=0:P3=0:P6=0:N5=0
1110 FOR I=1 TO 4:FOR J=15 TO 16:C(I,J)=0:NEXT:NEXT:FOR I=1 TO 37:D(I)=0:NEXT
1120 D(0)=4:D(10)=4:D(20)=4:D(30)=4:D(38)=4
1130 CLS:LOCATE11,6:PRINT"ｲﾏ ﾊﾟｲ ｦ ﾅﾗﾍﾞﾃ ｲﾏｽ";:LOCATE12,8:PRINT"ｼﾊﾞﾗｸ ｵﾏﾁ ｸﾀﾞｻｲ"
;
1140 '*****    ﾊｲﾊﾟｲ    *****
1150 FOR J1=1 TO 3:L=INT(RND(1)*6):IF 4>L THEN 1180
1160 A=4:P=4:GOSUB 1070:C(J1,13)=H:C(J1,15)=H:IF L=5 THEN1480
1170 GOTO 1190
1180 A=3:P=3:GOSUB 1070:C(J1,10)=H:C(J1,11)=H
1190 B=INT(RND(1)*(A+1)):IF B=0 THEN 1260
1200 FOR I=1 TO B
1210 H=INT(RND(1)*3)*10+INT(RND(1)*7)+1 :G=H
1220 IF D(G)=4 THEN1210
1230 G=G+1:IF G<=H+2 THEN1220
1240 FOR G=H TO H+2:D(G)=D(G)+1:NEXT:G=I*3:C(J1,G-2)=H:C(J1,G-1)=H+1:C(J1,G)=H+2
:NEXT I
1250 IF A=B THEN 1270
1260 P=2:FOR I=B+1 TO A:GOSUB 1070:G=I*3:C(J1,G-2)=H:C(J1,G-1)=H:C(J1,G)=H:NEXT
1270 ON L+1 GOTO 1280,1340,1410,1450,1490
1280 H=INT(RND(1)*3)*10+INT(RND(1)*6+2):IF D(H)=4 THEN 1280
1290 IF D(H+1)=4 THEN 1280
1300 C(J1,16)=H+2
1310 C(J1,15)=H-1
1320 D(H)=D(H)+1:D(H+1)=D(H+1)+1:C(J1,12)=H:C(J1,13)=H+1:NEXT J1:GOTO1510
1330 '
1340 IF .5>RND(1) THEN G=8:GOTO 1360
1350 G=1
1360 H=INT(RND(1)*3)*10+G:IF D(H)=4 THEN 1340
1370 IF D(H+1)=4 THEN 1340
1380 IF G=8 THEN 1310
1390 C(J1,15)=H+2:GOTO 1320
1400 '
1410 H=INT(RND(1)*3)*10+INT(RND(1)*7)+1:IF D(H)=4 THEN 1410
1420 IF D(H+2)=4 THEN 1410
1430 D(H)=D(H)+1:D(H+2)=D(H+2)+1:C(J1,12)=H:C(J1,13)=H+2:C(J1,15)=H+1:NEXT J1:GO
TO1510
1440 '
1450 P=3:GOSUB 1070:C(J1,12)=H:C(J1,13)=H:C(J1,15)=H:C(J1,16)=C(J1,10):NEXT J1
1460 GOTO 1510
1470 '
1480 P=3:FOR G=1 TO 11 STEP 2:GOSUB 1070:C(J1,G)=H:C(J1,G+1)=H:NEXT
1490 NEXT J1
1500 '
1510 LOCATE 12,11:PRINT"ｺﾚｶﾗ ｻｲ ｦ ﾌﾘﾏｽ";
1520 FOR I=1 TO 13
1530 H=INT(RND(1)*37)+1:IF D(H)=4 THEN 1530
1540 C(4,I)=H:D(H)=D(H)+1:NEXT
1550 'D(i) ｦ E(L) ﾆ ｲﾚﾙ
1560 L=0:FOR I=1 TO 37:G=4-D(I):IF G THEN FOR H=1 TO G:L=L+1:E(L)=I:NEXT
1570 NEXT I
1580 'ｻｲｺﾛﾌﾘ
1590 CLS:H=INT(RND(1)*6)+1:LOCATE 12,6:PRINT"ｻｲ  No.1.....";H;:G=INT(RND(1)*6)+1
:LOCATE 12,9:PRINT"ｻｲ  No.2.....";G;
1600 H3=((H+G-2)MOD4)+1:IF H3=1 THEN A=&H4B4C:GOTO1640:'■ﾍﾟｲ
1610 IF H3=2 THEN A=&H403E:GOTO 1640:'■ｼｬｰ
1620 IF H3=3 THEN A=&H466E:GOTO 1640:'■ﾅﾝ
1630                 A=&H456C:'■ﾄﾝ
1640 LOCATE 9,12:PRINT"ｱﾅﾀ ﾉ ﾊﾞｼｮ ﾊ ｢  ｣ ﾃﾞｽ":PRINT@(376,90),A;
1650 'E(L) ｦ ｷﾙ
1660 FOR I=1 TO 150:SWAP E(RND(1)*42+.5),E(RND(1)*42+42.5):NEXT
1670 'ﾘｰﾁ ｾﾝｹﾞﾝ ﾃｲｷﾞ
1680 FOR I=1 TO 3:P(I)=INT(RND(1)*2):Q(I)=RND(1)*65+8:NEXT
1690 CLS:FOR I=25 TO 177 STEP 32:LINE@(17,I)-(638,I+30),OR,4,BF:NEXT:LINE@(17,18
4)-(638,196),OR,4,BF
1700 L=0:N3=N3+1:H=N3 MOD 4
```

```
1710 IF H=1 THEN A=&H306C:GOTO1750
1720 IF H=2 THEN A=&H4673:GOTO1750
1730 IF H=3 THEN A=&H3B30:GOTO1750
1740 A=&H3B4D
1750 PRINT@(0,0),&H456C;A;&H3649;:LINE@(240,0)-(400,17),OR,2,BF:PRINT@(256,1),&H2346;&H234D;&H2338;&H237B;&H3B4D;&H3F4D;&H4B63;&H3F7D;:PRINT@(80,0),&H4C67;:PRINT@(100,0),&H4977;
1760 X=150:Y=0:E=35-H3:GOSUB160:G=H3*32-8:LINE@(0,G)-(15,G+31),OR,3,BF
1770 PRINT@(0,24),&H323C:PRINT@ &H3248:PRINT@ &H4250:PRINT@ &H4C4C:PRINT@ &H3E65:PRINT@ &H3248:PRINT@ &H3C2B:PRINT@ &H3248;:LOCATE 31,0:PRINT"ﾄﾞﾗﾊﾟｲ";:E=E(82):X=605:GOSUB160:LOCATE34,22:PRINT"ﾂﾓﾊﾟｲ";
1780 'ﾘｰﾊﾟｲ
1790 X=40:Y=154:FOR I=1 TO 12:E=C(4,I):FOR J=I+1 TO 13:IF E>C(4,J) THEN SWAP E,C(4,J)
1800 NEXT:C(4,I)=E:GOSUB160:X=X+34:NEXT:E=C(4,13):GOSUB160
1810 'A-N ｦ ﾃﾞｨｽﾌﾟﾚｰ ﾆ ﾀﾞｽ
1820 X=14:FOR I=1 TO 14:X=X+34:LINE@(X,176)-(X+15,183),PSET,1,BF:SYMBOL(X,176),R$(I),2,1:NEXT:J1=H3:IF J1=4 THEN760
1830 GOTO190
1840 LOCATE 9,14:PRINT"***  ﾃﾝｽｳ ｹｲｻﾝ  ***";
1850 LOCATE 13,15:INPUT"ｷｿ ﾉ ﾃﾝｽｳ ";A:Z3=A:LOCATE 14,16:INPUT"ﾅﾝﾊﾝ ﾃﾞｽｶ ";B:IF B THEN FOR I=1 TO B:Z3=Z3*2:NEXT
1860 Z3=INT(Z3/100+.995)*100:IF Z3>2000 THEN Z3=2000
1870 IF P1 THEN 1910
1880 IF H3-P2 THEN 1900
1890 Z3=Z3*2:F(P2)=F(P2)+Z3*3:F(P3)=F(P3)-Z3*3:GOTO1950
1900 F(P2)=F(P2)+Z3*4:F(P3)=F(P3)-Z3*4:GOTO1950
1910 IF H3-P1 THEN 1930
1920 Z3=Z3*2:F(J1)=F(J1)+Z3*4:GOTO1940
1930 F(J1)=F(J1)+Z3*5:F(H3)=F(H3)-Z3
1940 FOR I=1 TO 4:F(I)=F(I)-Z3:NEXT
1950 LOCATE 10,18:PRINT"***   ﾃﾝｽｳ ﾋｮｳ   ***";:LOCATE 10,19:PRINT"ﾀﾀﾞｲﾏﾉ ｱｶﾞﾘﾃﾝ  ";Z3;:FOR I=1 TO 4:LOCATE12,I+19:PRINT S$(I);SPC(6);F(I);:NEXT
1960 LOCATE 8,24:PRINT"ﾏﾀ ｻｲ ｦ ﾌｯﾃ ｸﾀﾞｻｲ。 Yes=Y";:GOSUB2170:IF (F$="y")+(F$="Y") THEN 1100
1970 CLS:LINE@(100,70)-(539,119),OR,1,BF:PRINT@(250,76),9522;8508;9568;9295;9504;15658;9322;9292;8483:PRINT@(170,96),19563;19520;9265;9310;9287;9294;8480;17211;9261;19506;9322;9330;8480;9312;9269;9308;9323;9305;9271;8483:END
1980 'ﾁｲ ﾎﾟﾝ ｱｸｼｮﾝ
1990 LINE@(X,Y)-(X+32,Y+20),XOR,2,BF:'▬ﾊﾟｲ ﾉ color ｦ ｶｴﾙ
2000 H=N5+1:G(H)=E(L):G=L MOD 4:L=L+1:IF J1=1 AND G THEN L=L+2
2010 J1=4:G=1:GOSUB 2030:GOSUB 2030:N5=N5+1:Y=154:GOSUB 2110:GOTO870
2020 BEEP 1:FOR I=1 TO 400:NEXT:BEEP 0
2030 LOCATE 2,18:PRINT "ﾜｲﾊｲ";-G;:LOCATE8,18:GOSUB2170
2040 FOR I1=1 TO 28:IF F$=R$(I1) THEN 2060
2050 NEXT:GOTO2020
2060 IF I1>14 THEN I1=I1-14
2070 IF I1>13-N5 THEN 2020
2080 LOCATE2,18:PRINT SPC(6);:H=H+1:G=G+1:J1=4:G(H)=C(4,I1)
2090 FOR J=I1 TO 13:C(4,J)=C(4,J+1):NEXT
2100 N5=N5+1:GOTO440
2110 FOR I=1 TO N5-1:FOR J=I+1 TO N5:IF G(I)>G(J) THEN SWAP G(I),G(J)
2120 NEXT:NEXT:X=524:GOSUB2150
2130 X=492-N5*34:FOR I=1 TO N5:E=G(I):X=X+34:GOSUB160:NEXT:RETURN
2140 '
2150 LINE@(X,154)-(526-N5*34,174),PSET,4,BF:RETURN
2160 '▬INKEY$ yellow cursor
2170 PRINT"?";:X1=POS(0):Y1=CSRLIN:COLOR6
2180 LOCATE X1,Y1:PRINT"■";:FOR I=1 TO 310:NEXT:LOCATE X1,Y1:PRINT" ";:F$=INKEY$:IF F$<>"" THEN2200
2190 FOR I=1 TO 520:NEXT:F$=INKEY$:IF F$="" THEN2180
2200 COLOR7:RETURN
4001 X1=X+16:Y1=Y+10:R1=15:CL=0:GOSUB4030:R1=12:CL=4:GOSUB4030:R1=6:CL=2:GOTO4020
4002 X1=X+16:Y1=Y+5:CL=0:R1=9:GOSUB4030:CL=4:R1=6:GOSUB4020:GET@A(X,Y+1)-(X+29,Y+9),ZB,G:PUT@A(X,Y+11)-(X+29,Y+19),ZB,PSET:RETURN
4003 X1=X+8:Y1=Y+4:CL=4:R1=4:GOSUB4020:CL=0:R1=7:GOSUB4030:GET@A(X,Y+2)-(X+14,Y+8),Z8,G:PUT@A(X+17,Y+14)-(X+31,Y+20),Z8,PSET:X1=X+16:Y1=Y+10:GOSUB4030:CL=2:R1=4:GOSUB4020:GET@A(X+9,Y+8)-(X+23,Y+13),Z7,G:RETURN
4004 PUT@A(X+1,Y+3)-(X+15,Y+9),Z8,PSET:PUT@A(X+16,Y+3)-(X+30,Y+9),Z8,PSET:GET@A(X+1,Y+3)-(X+30,Y+9),Z9,G:PUT@A(X+1,Y+12)-(X+30,Y+18),Z9,PSET:RETURN
4005 PUT@A(X+1,Y+2)-(X+30,Y+8),Z9,PSET:PUT@A(X+1,Y+14)-(X+30,Y+20),Z9,PSET:PUT@A(X+9,Y+8)-(X+23,Y+13),Z7,PSET:RETURN
4006 PUT@A(X+1,Y+2)-(X+30,Y+8),Z9,PSET:PUT@A(X+2,Y+10)-(X+16,Y+15),Z7,PSET:PUT@A(X+2,Y+14)-(X+16,Y+19),Z7,PSET:GET@A(X+3,100)-(X+16,110),ZB,G:PUT@A(X+17,100)-(X+30,110),ZB,PSET:RETURN
4007 PUT@A(X,Y+1)-(X+14,Y+7),Z8,PSET:PUT@A(X+17,Y+5)-(X+31,Y+11),Z8,PSET:R1=4:CL=4:X1=X+16:Y1=Y+5:GOSUB4020:R1=7:CL=0:GOSUB4030:PUT@A(X+3,100)-(X+16,110),ZB,PSET:PUT@A(X+17,100)-(X+30,110),ZB,PSET:RETURN
4008 FORY1=Y+2 TO Y+16 STEP 4:PUT@A(X+1,Y1)-(X+30,Y1+6),Z9,PSET:NEXT:RETURN
```

プログラム・リスト

```
4009 X1=X+6:Y1=Y+4:R1=6:CL=0:GOSUB4030:R1=3:CL=4:GOSUB4020:Y1=Y+16:GOSUB4020:R1=
6:CL=0:GOSUB4030:Y1=Y+10:GOSUB4030:R1=4:CL=2:GOSUB4020:GET@A(X+1,90)-(X+12,110),
ZB,G:PUT@A(X+11,90)-(X+22,110),ZB,PSET:PUT@A(X+21,90)-(X+32,110),ZB,PSET:RETURN
4020 CIRCLE@(X1,Y1),R1,CL:PAINT(X1,Y1),CL,CL:RETURN
4030 CIRCLE@(X1,Y1),R1,CL:RETURN
4100 DEFINT A-E,G-Z:A=0:B=0:CL=0:E=0:F$="":G=0:H=0:H3=0:I=0:I1=0:J=0:J1=0:J3=0:L
=0:N3=0:N5=0:N7=0:P=0:P1=0:P2=0:P3=0:P6=0:R1=0:X=0:X1=0:X2=0:Y=0:Y1=0:Y2=0:Z3=0:
KEY 1,"  ":KEY 2,"ﾘｰﾁ"+CHR$(13)
4110 DIM C(4,16),D(38),E(84),F(4),G(13),P(4),Q(4),R$(28),S$(4),Z1(77),Z2(81),Z21
(127),Z29(81),Z3(114),Z4(90),Z5(114),Z6(114),Z7(114),Z8(114),Z9(119),ZA(1744),ZB
(131)
4120 COLOR7,0:WIDTH 40,25:LINE@(118,47)-(524,162),OR,4,BF
4130 LINE@(240,51)-(400,68),XOR,6,BF:PRINT@(256,52),9030;9037;9016;9083;15181;16
205;19299;16253
4140 X=134:Y=90:FOR I=1 TO 9:X=X+34:LINE@(X,90)-(X+32,110),OR,7,BF
4150 ON I GOSUB 4001,4002,4003,4004,4005,4006,4007,4008,4009
4160 NEXT I
4170 X=168:Y=125:GOSUB4370:FOR X1=X+47 TO X+71 STEP 8:FOR Y1=Y+2 TO Y+4 STEP 2:L
INE@(X1,Y1)-(X1+4,Y1),PSET,0,B:NEXT:NEXT
4180 Y=135:GOSUB4370:X1=X+60:Y1=Y+3:CIRCLE@(X1,Y1),4,2:PAINT(X1,Y1),2,2:Y=145:GO
SUB4370:X1=X+60:Y1=Y+3:CIRCLE@(X1,Y1),4,2:PAINT(X1,Y1),2,2
4190 FOR X1=X+50 TO X+70 STEP 20:FOR Y1=Y+2 TO Y+4 STEP 2:CIRCLE@(X1,Y1),3,2,,,,
F:NEXT:NEXT:LINE@(400,135)-(430,147),PSET,7,BF:CIRCLE@(415,141),7,2:PAINT(415,14
1),2,2
4200 GET@A(169,94)-(199,106),Z1,G:GET@A(206,92)-(230,108),Z2,G
4210 GET@A(236,90)-(267,108),Z3,G:GET@A(270,93)-(301,107),Z4,G
4220 GET@A(304,90)-(335,108),Z5,G:GET@A(338,90)-(369,108),Z6,G
4230 GET@A(372,90)-(403,108),Z7,G:GET@A(406,90)-(437,108),Z8,G
4240 GET@A(440,90)-(472,108),Z9,G
4250 RESTORE4390:FOR I=1 TO 28:READ R$(I):NEXT:S$(1)="ｼﾓﾁｬ":S$(2)="ﾄｲﾒﾝ":S$(3)="
ｶﾐﾁｬ":S$(4)="ｱﾅﾀ "
4260 FOR I=1 TO 4:F(I)=27000:NEXT
4270 RANDOMIZE(TIME MOD 32767)
4280 CLS:LOCATE 8,5:PRINT"***      FM 4 ﾏｰｼﾞｬﾝ     ***";
4290 LOCATE1,7:PRINT"ｱﾅﾀ ｦ ｺﾚｶﾗ ｽﾊﾞﾗｼｲ ﾌﾟﾛ ﾉ ﾏｰｼﾞｬﾝ ﾉ ｾｶｲ ﾍ ｺﾞｱﾝﾅｲ ｼﾏｽ";
4300 LOCATE0,11:PRINT"Personal computer FM-8 ｶﾞ ｱﾅﾀ ﾉ ｶﾐﾁｬ, ﾄｲﾒﾝ, ｼﾓﾁｬ ｦ ｳｹﾓﾁﾏｽ"
;
4310 LOCATE0,14:PRINT"ｱﾅﾀ ﾉ ﾖｳｷｭｳ ｽﾙ FM-8 ﾉ ﾂﾖｻ ｦ ﾆｭｳﾘｮｸ ｼﾃ ｸﾀﾞｻｲ";
4320 LOCATE4,17:PRINT"ｻｲｺｳ ｸﾗｽ    ﾌﾟﾛ ｶﾞ ｵｱｲﾃ ｼﾏｽ....1";
4330 LOCATE4,18:PRINT"ﾍｲｷﾝ ｸﾗｽ   ｲｯﾊﾟﾝ ｱﾏﾁｭｱ ﾃｲﾄﾞ...2";
4340 LOCATE4,19:PRINT"ﾆｭｳﾓﾝ ｸﾗｽ  ｼｮｼﾝｼｬ ｶﾞ ｵｱｲﾃ....3";:GOSUB2170
4350 N7=ASC(F$)-48:IF (N7>3)+(N7<1) THEN4280
4360 GOTO1120
4370 LINE@(X,Y)-(X+120,Y+6),PSET,7,BF:RETURN:'▬CRT ﾉ color ｦ ｶｴﾙ
4380 FOR I=1 TO 7:LINE@(0,0)-(639,199),XOR,I,BF:NEXT:RETURN
4390 DATA a,b,c,d,e,f,g,h,i,j,k,l,m,n,A,B,C,D,E,F,G,H,I,J,K,L,M,N
```



FM-8 SPACE INVADERS

■COMPAC O.H.

『FM-8では高速ゲームはできない』と言われてきましたが，ここに紹介するのはまさしくその『高速ゲーム』です．題して『高速スペース・インベーダー・ゲーム』．詳細を紹介できないのが残念ですが，実力のある方は解析して他のゲームで挑戦してみてください．

とにかくリストの通りにプログラムを入力して，一応テープにSAVEしておいてください．スタート番地は5000(16)番地です．砲台の移動は[1]，[3]キーでそれぞれ左，右に動きます．移動の停止は[2]のキーです．ビームの発射は[STOP]キーで，再ゲームは[SPACE]キーです．ゲームの内容はあまりにも有名なので，特に説明はしませんが，スピードも色もまあまあのできではないかと思います．

◘参考文献

1) I/O，'79年7月号，近藤洋一他，〝スペース・インベーダ〟
2) I/O，'81年10月号，FALCON・S，〝SPACE INVADERS〟

写真1

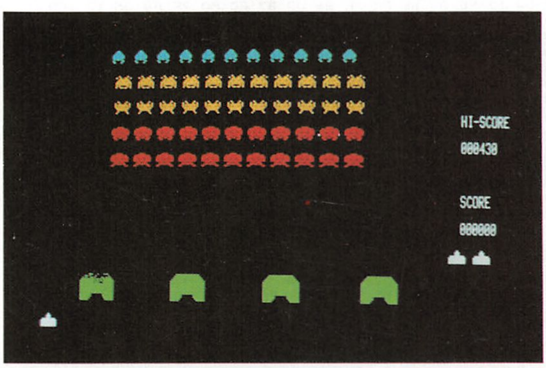
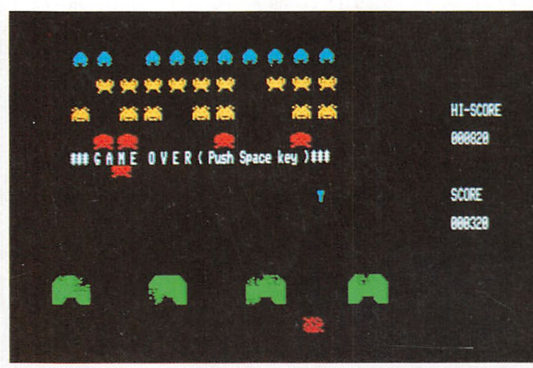


写真2



スペース・インベーダー ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5000 10 CE 6F FF 86 20 1F 8B 17 0C 1C 8D 2B 8D 64 17 :9B
5010 04 EC 17 04 4A 17 06 75 17 00 E1 17 04 D8 17 05 :EE
5020 F6 17 03 6B 17 05 F0 17 02 E2 17 06 BE 17 07 6A :E5
5030 17 08 2B 17 09 DC 20 DA 8E 00 00 9F 00 8E 20 02 :1D
5040 6F 80 8C 40 00 26 F9 86 03 97 09 8E BC 00 9F 0A :F6
5050 86 37 97 0F 8E 87 80 9F 10 9F 12 8E 37 80 9F 8A :C6
5060 8E 30 F0 9F 90 9F 92 30 8D 04 13 BF 01 E1 86 E9 :F2
5070 97 1A 39 9E 90 C6 0B 86 E3 A7 84 A7 01 A7 02 86 :54
5080 E2 A7 89 00 A0 A7 89 00 A1 A7 89 00 A2 A7 89 01 :86
5090 40 A7 89 01 41 A7 89 01 42 86 E1 A7 89 01 E0 A7 :44
50A0 89 01 E1 A7 89 01 E2 A7 89 02 80 A7 89 02 81 A7 :8A
50B0 89 02 82 30 04 5A 26 BF 8E 36 97 C6 04 86 03 CE :FC
50C0 04 04 34 10 A7 80 EF 81 EF 81 A7 80 30 88 4A EF :6B
50D0 81 A7 80 A7 80 EF 81 35 10 30 88 10 5A 26 E3 9E :4D
50E0 8A 86 55 A7 80 A7 80 A7 84 39 96 02 D6 02 9B 02 :24
50F0 97 02 9B 02 D7 02 9B 02 4C 97 02 39 C6 0A 96 0F :3F
---------------------------------------------------------
Sum: 15 5E 19 49 8A EB F0 92 0A B5 0E AA C0 FC B3 46 :F8


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5100 5A 49 2A FC 86 80 49 5A 26 FC 97 0C 5F 96 0F 5C :97
5110 49 2A FC 5A 86 FF 44 5A 26 FC 97 8C 94 04 91 0C :66
5120 27 01 39 96 8E 10 26 01 34 96 0D 10 2B 00 8D 96 :F1
5130 8D 26 0A 8E 31 31 17 01 0A 10 2B 01 20 96 8D 26 :74
5140 06 9E 10 30 01 9F 10 C6 02 3D 86 50 3D D3 90 1F :2E
5150 01 34 10 30 88 2B C6 0B A6 84 81 DD 26 14 A6 1F :80
5160 84 E0 81 E0 26 0C A6 1F 34 10 30 1E 9F 80 35 10 :B2
5170 97 82 A6 1F 81 E1 25 08 81 E4 24 04 A7 84 6F 1D :B1
5180 30 1C 5A 26 D3 35 10 30 01 17 02 B2 9E 10 96 0E :32
5190 26 04 C6 5F 20 02 C6 51 F7 FC 80 BF FC FE 96 8D :D7
51A0 B7 FC FB 7F FC FA 7F FD 05 0C 8D 96 8D 81 05 26 :0C
51B0 08 0F 8D 9E 90 30 01 9F 90 03 0E 39 96 8D 26 09 :CE
51C0 8E 30 F0 8D 7E 10 2B 00 94 96 8D 26 06 9E 10 30 :B5
51D0 1F 9F 10 C6 02 3D 86 50 3D D3 90 1F 01 34 10 30 :DD
51E0 1F C6 0B A6 84 81 DD 26 0E A6 01 84 E0 81 E0 26 :3E
51F0 06 A6 01 97 82 9F 80 A6 01 81 E1 25 08 81 E4 24 :A4
---------------------------------------------------------
Sum: 60 34 64 0B 00 45 CF E7 54 05 DD 26 93 0B CF 03 :CA


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5200 04 A7 84 6F 03 30 04 5A 26 D9 35 10 30 1F 17 02 :DB
5210 2D 9E 10 96 0E 26 04 C6 5F 20 02 C6 51 F7 FC 80 :7A
5220 BF FC FE 96 8D B7 FC FB 86 FF B7 FC FA 7F FD 05 :3D
5230 0C 8D 96 8D 81 05 26 08 0F 8D 9E 90 30 1F 9F 90 :B8
5240 03 0E 39 C6 16 A6 84 81 E1 25 09 81 E4 24 05 86 :F4
5250 FF 97 8E 39 5A 26 01 39 30 88 50 20 E8 96 8D 26 :70
5260 0C 86 05 97 8D 9E 10 30 89 02 80 9F 10 4A C6 02 :65
5270 3D 86 50 3D D3 90 1F 01 34 10 C6 0B A6 84 84 E0 :76
5280 81 E0 26 3B A6 88 50 81 DD 27 0E A6 88 51 81 DD :B0
5290 27 07 A6 88 52 81 DD 26 0D A6 84 97 82 34 10 30 :F6
52A0 88 50 9F 80 35 10 A6 84 81 E1 25 13 81 E4 24 0F :98
52B0 A7 88 50 A7 88 51 A7 88 52 6F 84 6F 01 6F 02 30 :84
52C0 04 5A 26 B8 35 10 30 88 50 17 01 72 96 0E 2B 04 :E6
52D0 C6 4F 20 02 C6 41 9E 10 96 8D B7 FC FB F7 FC 80 :30
52E0 BF FC FE 7F FD 05 0A 8D 26 21 0F 8E 9E 90 30 88 :9B
52F0 50 9F 90 03 0D 03 0E 8E 37 80 86 43 E6 80 C4 E0 :B8
---------------------------------------------------------
Sum: F7 82 D3 21 A9 CF 3E 74 E8 A6 B3 AB CE 29 5D DD :B4


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5300 C1 E0 26 04 C6 FF D7 03 4A 26 F1 39 96 04 84 3F :61
5310 81 20 27 01 39 17 01 C3 C1 31 27 05 C1 33 27 3A :50
5320 39 9E 8A 8C 37 80 26 01 39 A6 1F 27 04 86 FF 97 :10
5330 14 A6 02 A7 1F 6F 02 30 1F 9F 8A C6 81 9E 0A 30 :8A
5340 1F 9F 0A 17 02 BE 86 81 B7 FC 80 BF FC FE 86 FF :17
5350 B7 FC FA 7F FD 05 7F FD 05 39 9E 8A 8C 37 C0 26 :B9
```

スペース・インベーダー ダンプ・リスト

```
5360 01 39 A6 03 27 04 86 FF 97 14 A6 84 A7 03 6F 84 :05
5370 30 01 9F 8A 9E 0A 30 01 9F 0A 17 02 87 86 81 B7 :3A
5380 FC 80 BF FC FE 7F FC FA 7F FD 05 7F FD 05 39 96 :7B
5390 16 2B 03 16 00 DB 96 04 84 3F 81 20 27 01 39 9E :32
53A0 97 A6 88 B0 27 53 84 E0 81 E0 10 27 00 7A A6 88 :93
53B0 B0 81 22 27 6C 84 A0 81 A0 27 27 0F 16 6F 9F 20 :CC
53C0 97 30 88 B0 17 02 3D A6 84 B7 FC FC 6A 84 17 02 :35
53D0 D5 BF FC FE 86 FF B7 FC FD 86 01 B7 FC 80 7F FD :F9
53E0 05 39 17 02 1F 86 90 B7 FC 80 9E 17 BF FC FE 7F :AC
53F0 FD 05 0F 16 6F 9F 20 97 39 A6 84 A7 88 B0 6F 84 :21
-------------------------------------------------------
Sum: 5D 18 38 0A D5 2D 15 C4 2F 95 78 40 79 B8 A4 7E :61
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5400 30 88 B0 9F 97 8C 30 50 25 DB 9E 17 30 89 FD 80 :92
5410 9F 17 17 01 EF BF FC FE 86 11 B7 FC 80 7F FD 05 :C1
5420 39 8D BF 9E 9F 16 05 2B A6 88 B0 97 82 30 88 B0 :67
5430 A6 82 84 E0 81 E0 27 F8 30 01 9F 80 20 A4 17 01 :38
5440 C3 CE FC 81 C6 0B 34 04 5F A6 84 81 E1 25 06 81 :AE
5450 E4 24 02 C6 FF E7 C0 30 04 35 04 5A 26 E8 39 0C :90
5460 04 17 FC 86 96 05 4C 97 05 81 10 27 01 39 0F 05 :26
5470 39 0F 19 1C AF 1A 50 96 19 26 02 0F 99 39 03 19 :6A
5480 96 99 26 35 9E 8A 30 88 B1 9F 97 A6 84 26 31 86 :58
5490 FF 97 16 86 DD A7 84 9E 0A 30 89 FD 81 9F 17 17 :E6
54A0 01 62 86 10 BF FC FE B7 FC 80 7F FD 05 0C 1A 03 :8F
54B0 99 96 1A 81 0F 26 02 0F 1A 86 50 AA E4 A7 E4 3B :54
54C0 84 E0 81 E0 26 E7 30 1F A6 84 84 E0 81 E0 27 F6 :2D
54D0 30 01 9F 80 8E 20 97 9F 97 20 D2 17 01 26 86 20 :A1
54E0 B7 FC 80 7F FD 05 7F FD 05 17 01 18 F6 FC F9 7F :CF
54F0 FD 05 7F FD 05 39 8E 00 19 30 1F 26 FC 39 17 01 :25
-------------------------------------------------------
Sum: 29 D0 18 2F AF EA 70 79 2E B4 A3 BA 55 0E ED 52 :A3
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5500 03 86 30 B7 FC 80 7F FD 05 17 00 C7 17 00 F5 9E :F5
5510 0A BF FC FE 86 81 B7 FC 80 7F FD 05 86 FF 97 0E :A8
5520 8D 78 17 00 DF 9E 06 BF FC FC 8E AA C7 BF FC FE :0E
5530 86 03 B7 FC 80 7F FD 05 17 00 C9 30 8D 00 4C CE :F4
5540 FC 81 A6 80 A7 C0 26 FA 8E A5 C7 BF FC FE 86 04 :67
5550 B7 FC 80 7F FD 05 17 00 AB 9E 00 BF FC FC 8E 9B :F4
5560 C7 BF FC FE 86 03 B7 FC 80 7F FD 05 17 00 95 30 :99
5570 8D 00 1E CE FC 81 A6 80 A7 C0 26 FA 8E 96 C7 BF :4D
5580 FC FE 86 04 B7 FC 80 7F FD 05 39 53 43 4F 52 45 :ED
5590 00 48 49 2D 53 43 4F 52 45 00 8D 68 86 20 C6 0B :A6
55A0 8E FC 81 A7 80 5A 26 FB 6F 84 86 04 B7 FC 80 8E :EB
55B0 AF C5 BF FC FE 7F FD 05 D6 09 8E AF C5 5A 27 11 :21
55C0 8D 42 BF FC FE 86 81 B7 FC 80 7F FD 05 30 04 20 :97
55D0 EC 39 39 C6 04 34 04 86 02 3D 86 50 3D D3 90 1F :BA
55E0 01 17 FE 5A 9E 10 BF FC FE 86 51 B7 FC 80 35 04 :1A
55F0 F7 FC FB 7F FD 05 5A 2A DC 0F 8D 0F 8E 39 7F FD :BD
-------------------------------------------------------
Sum: D1 91 3A EB 2C 4E 63 67 57 F8 FB A4 9F CF 4B 35 :A7
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5600 05 7F FD 05 B6 FD 05 2B F5 86 80 B7 FD 05 B7 FD :D1
5610 05 B6 FC 80 26 E8 39 9E 80 26 01 39 6F 84 6F 01 :5F
5620 6F 02 17 00 81 9F 83 8D DB 86 31 B7 FC 80 BF FC :38
5630 FE 0A 0F 0F 16 6F 9F 20 97 7F FD 05 8E 00 50 C6 :26
5640 20 86 41 8D 0C C6 20 86 40 8D 06 30 1F 26 F0 20 :44
5650 07 B7 FD 03 5A 26 FD 39 8E 00 00 9F 80 D6 82 C4 :3D
5660 03 9E 06 3A 9F 06 8C 00 96 25 0B 96 08 26 07 0C :AF
5670 09 0A 08 17 FF 24 17 FF 8B 9E 06 BF FC FC 8E AA :89
5680 C7 BF FC FE 86 03 B7 FC 80 7F FD 05 39 9E 83 26 :3D
5690 01 39 CE 00 00 DF 83 17 FF 6A 86 9F B7 FC 80 BF :01
56A0 FC FE 7F FD 05 39 30 89 D0 00 CE 80 00 1F 10 8E :48
56B0 00 50 9F B9 33 C9 02 80 93 B9 24 F8 33 C9 FD 80 :07
56C0 D3 B9 DF B9 D3 B9 1F 01 39 C6 16 A6 84 97 1B 84 :45
56D0 E0 81 E0 27 0B 30 88 B0 5A 26 F0 1C FE 39 30 1F :ED
56E0 A6 1F 84 E0 81 E0 27 F6 1A 01 39 96 04 84 3F 81 :D9
56F0 20 27 01 39 8E 20 28 C6 03 A6 84 27 02 8D 06 30 :36
-------------------------------------------------------
Sum: E7 EC 97 22 22 D6 82 BD 68 36 FE 6B 44 8A DC A1 :15
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5700 05 5A 26 F5 39 34 14 10 AE 89 00 81 10 8C 37 7F :15
5710 10 24 01 2B A6 A8 50 26 2D A6 A4 A7 A8 50 6F A4 :4D
5720 31 A8 50 10 AF 89 00 81 10 AE 01 31 A9 02 80 10 :1D
5730 AF 01 17 FE CF 10 BF FC FE A6 03 B7 FC 80 7F FD :B5
5740 05 7F FD 05 35 94 81 DD 27 36 81 01 25 2A 81 05 :61
5750 24 26 6F 84 6F A4 1F 21 30 88 50 17 FE A6 A6 84 :7D
5760 6A 84 B7 FC FC 17 FF 3E BF FC FE 7F FC FD 86 01 :A9
5770 B7 FC 80 7F FD 05 35 94 81 55 26 04 86 FF 97 14 :AD
5780 17 FE 81 86 90 EE 01 33 C9 02 80 6F A4 FF FC FE :25
5790 B7 FC 80 7F FD 05 6F 84 35 94 96 8D 27 01 39 96 :8A
57A0 04 84 0F 81 04 27 01 39 8E 20 28 C6 03 A6 84 26 :6C
57B0 03 8D 07 39 30 05 5A 26 F4 39 34 10 9E 8A 17 FF :34
57C0 08 25 1D 17 F9 24 8A 30 84 7F 1F 89 86 37 1F 01 :C0
57D0 17 FE F6 25 02 35 90 17 F9 10 8D B4 25 F7 20 15 :9C
57E0 96 0F 81 03 25 F1 49 49 8A 7D 34 04 17 F8 FB A0 :BA
57F0 E0 25 E2 20 E2 35 20 34 20 30 89 00 A1 8C 37 80 :2F
-------------------------------------------------------
Sum: A9 AE BE 50 BD 67 45 5D 27 BD 6B BE D1 0C 2A BD :FC
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5800 24 D3 A6 84 81 DD 27 CD 81 55 26 06 86 FF 97 03 :94
5810 35 90 86 FF A7 A0 AF A9 00 80 30 88 B0 17 FE 86 :6C
5820 AF A4 17 FD DF BF FC FE 96 1B 84 AF A7 22 AE A9 :03
5830 00 80 A7 84 B7 FC 80 7F FD 05 7F FD 05 35 90 17 :BC
5840 FD C2 EE 01 33 C9 02 80 FF FC FE 86 90 B7 FC 80 :6E
5850 7F FD 05 7F FD 05 6F 84 6F 99 00 81 35 94 96 26 :03
5860 10 26 01 77 96 1D 26 70 96 04 81 F8 27 01 39 96 :01
5870 0F 81 08 25 F9 17 F8 72 80 FD 25 F2 17 F8 6B 84 :C9
5880 01 81 01 27 0A 8E 80 3E CE 30 3E C6 FF 20 07 5F :87
5890 8E 80 00 CE 30 00 A6 C4 26 D4 A6 41 26 D0 A6 42 :35
58A0 26 CE A6 43 26 C8 A6 44 26 C4 A6 45 26 C0 0A 1D :95
58B0 D7 1E 9F 1F DF 9F 17 FD 4B BF FC FE F7 FC FA 86 :BC
58C0 05 B7 FC 80 96 21 B7 FC FB 7F FD 05 03 21 8E 22 :F2
58D0 22 AF C1 AF C1 AF C4 39 96 04 84 3F 81 20 27 01 :D4
58E0 39 96 1E 26 36 9E 9F 8C 30 3E 10 27 00 F2 A6 06 :55
58F0 26 61 A6 84 6F 84 A7 06 30 01 9F 9F 9E 1F 30 01 :AE
-------------------------------------------------------
Sum: B5 35 AD 50 B8 21 85 E3 EE D4 B3 7F 49 AF 45 77 :D0
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5900 9F 1F 17 FC FF BF FC FE 7F FC FA 96 21 B7 FC FB :63
5910 86 05 B7 FC 80 7F FD 05 03 21 39 9E 9F 8C 30 00 :95
5920 10 27 00 BC A6 1F 26 2B A6 84 A7 1F 6F 05 30 1F :BC
5930 9F 9F 9E 1F 30 1F 17 FC CB BF FC FE 9F 1F 86 FF :24
5940 B7 FC FA 96 21 B7 FC FB 86 05 B7 FC 80 7F FD 05 :51
5950 03 21 39 6F 80 6F 80 6F 80 6F 80 6F 80 6F 84 9E :99
5960 1F 17 FC A0 BF FC FE 86 04 B7 FC 80 96 1A 26 17 :35
5970 8E 20 33 BF FC 81 8E 30 30 BF FC 83 8E 20 20 BF :D6
5980 FC 85 7F FC 87 20 34 8E 20 20 17 F7 5D 2B 0B 8E :D4
5990 20 31 17 F7 55 2B 03 8E 20 32 BF FC 81 B6 FC 82 :32
59A0 81 20 27 08 8E 30 30 17 F7 40 2B 03 8E 35 30 BF :EC
59B0 FC 83 8E 20 20 BF FC 85 7F FC 87 0A 26 0F 1D B6 :A1
59C0 FC 82 5F 84 03 27 05 CB 0A 4A 26 FB B6 FC 83 84 :89
59D0 0F 34 02 EB E0 7F FD 05 16 FC 86 0C 27 27 0F 39 :CB
59E0 9E 9F 6F 80 6F 80 6F 80 6F 80 6F 80 6F 84 0F 26 :10
59F0 17 FC 11 9E 1F BF FC FE 8E 20 20 BF FC 81 BF FC :5F
-------------------------------------------------------
Sum: 94 E8 FA DF AC 3E 0E 50 00 BE C8 05 CC DC 5D F6 :23
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5A00 83 BF FC 85 7F FC 87 86 04 B7 FC 80 7F FD 05 0F :12
5A10 1D 39 96 14 26 0D 96 03 10 26 00 B6 96 0F 10 27 :94
5A20 01 39 39 10 CE 6F FF 0F 14 0A 09 10 27 00 A5 9E :6F
5A30 0A 17 FB D0 BF FC FE 86 02 B7 FC 80 7F FD 05 17 :F8
5A40 01 8A 8E 20 28 C6 03 A6 84 27 1A 6F 84 6F 99 00 :90
5A50 81 17 FB B0 EE 01 33 C9 02 80 FF FC FE 86 90 B7 :76
5A60 FC 80 7F FD 05 30 05 5A 26 DD 96 16 27 16 0F 16 :9D
5A70 6F 9F 20 97 17 FB 8D 9E 17 BF FC FE 86 90 B7 FC :9B
5A80 80 7F FD 05 9E 8A 6F 80 6F 80 6F 84 9E 0A 17 FB :B4
5A90 73 BF FC FE 86 9F B7 FC 80 7F FD 05 17 FB 65 8E :0A
5AA0 BC 00 BF FC FE 9F 0A 86 81 B7 FC 80 7F FD 05 8E :67
5AB0 37 80 9F 8A 86 55 A7 80 A7 80 A7 84 17 FA DB 96 :B6
5AC0 8D 27 0C 17 F6 5D 20 F7 8E 03 E8 30 1F 26 FC 17 :42
5AD0 F5 40 35 10 9E 06 9C 00 25 16 17 FB 27 BF FC FC :E5
5AE0 9F 00 8E 9B C7 BF FC FE 86 03 B7 FC 80 7F FD 05 :85
5AF0 17 FB 11 30 8D 00 39 CE FC 81 A6 80 A7 C0 26 FA :11
-------------------------------------------------------
Sum: B6 28 25 58 F4 A5 AA CA 39 B4 17 79 A2 C4 25 73 :E3
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5B00 86 04 B7 FC 80 8E 9E AA BF FC FE 7F FD 05 9E 0A :75
5B10 17 FA F1 BF FC FE 86 02 B7 FC 80 7F FD 05 17 00 :0E
5B20 AB 17 F9 B7 C1 20 26 F9 86 03 17 F5 10 16 F4 DD :FE
5B30 2A 2A 2A 20 47 20 41 20 4D 20 45 20 20 4F 20 56 :1D
5B40 20 45 20 52 20 28 20 50 75 73 68 20 53 70 61 63 :86
5B50 65 20 6B 65 79 20 29 2A 2A 2A 00 35 10 8E 20 14 :9C
5B60 6F 80 8C 20 85 26 F9 0F 8D 0F 8E 0F 0D 86 37 97 :E8
5B70 0F 9E 92 8C 30 F0 26 14 30 89 00 A0 9F 92 9F 90 :DE
5B80 9E 12 30 89 05 00 9F 12 9F 10 20 16 8C 33 C0 27 :AA
5B90 13 30 88 50 9F 92 9F 90 9E 12 30 89 02 80 9F 12 :17
5BA0 9F 10 20 0E 8E 87 80 9F 10 9F 12 8E 30 F0 9F 90 :AF
5BB0 9F 92 8E BC 00 9F 0A 8E 37 80 9F 8A 8E 30 00 6F :BF
5BC0 80 8C 40 00 26 F9 17 FA 0A 16 F4 41 86 18 4A 27 :E0
5BD0 55 34 02 8D 04 35 02 20 F5 33 8D 00 33 8E 00 0F :F8
5BE0 27 15 E6 C0 27 11 4F 1F 02 1F 10 86 40 8D 1A 30 :56
5BF0 1F 27 EA 8D 03 20 E6 39 1F 20 86 41 8D 0B 1F 20 :DC
-------------------------------------------------------
Sum: 7F A2 EC 72 58 41 09 A3 49 19 E8 D6 0B 96 A1 99 :BF
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5C00 86 40 8D 05 30 1F 26 F0 39 B7 FD 03 5A 26 FD 39 :63
5C10 80 70 60 50 40 30 20 10 20 30 40 50 60 50 40 30 :40
5C20 20 50 40 30 20 00 39 B6 FD 05 2B FB 1A 50 86 80 :87
5C30 B7 FD 05 8D 74 7F FD 05 8E 00 00 30 1F 26 FC B6 :F0
5C40 FD 05 2B FB 86 80 B7 FD 05 17 00 91 7F FD 05 8E :9E
5C50 00 00 30 1F 26 FC B6 FD 05 2B FB 33 8D 00 D2 30 :11
5C60 8D 09 59 BF 20 00 C6 FF 86 80 B7 FD 05 B7 FD 05 :0B
5C70 A6 C0 B7 FC 81 F7 FC 80 7F FD 05 7F FD 05 B6 FD :C2
5C80 05 2B FB 86 80 B7 FD 05 B6 FC 80 26 E8 :E3
5C90 11 B3 20 00 26 D2 86 01 B7 FC 80 7F FD 05 7F FD :93
5CA0 05 8E 00 00 30 1F 26 FC 39 10 8E FC 80 30 8D 00 :14
5CB0 0A C6 21 A6 80 A7 A0 5A 26 F9 39 00 00 3F 59 41 :E9
5CC0 4D 41 55 43 48 49 93 D3 8F 90 34 03 1A 50 B7 D4 :68
5CD0 0A B6 D4 08 B6 D4 09 B6 D4 0A 35 83 00 10 8E FC :15
5CE0 80 30 8D 00 0A C6 41 A6 80 A7 A0 5A 26 F9 39 00 :6D
5CF0 00 3F 59 41 4D 41 55 43 48 49 91 D3 96 3E 81 00 :49
-------------------------------------------------------
Sum: 09 63 E8 9F FC B4 26 02 EB 37 05 A2 50 30 D3 55 :3C
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5D00 2C 93 3E 81 90 CE C0 00 7F D3 80 7F D3 80 7F D3 :92
5D10 80 7F D3 80 B6 D4 0A B6 D3 80 27 F8 81 01 27 0D :C4
```

スペース・インベーダー ダンプ・リスト

```
5D20 B7 D4 0A B7 D4 0A B6 D3 81 A7 C0 20 DB 7E C0 00 :D4
5D30 00 7F D3 80 10 CE CF FF 7F D3 80 7F D3 80 B6 D4 :AC
5D40 0A B6 D3 80 27 F8 B7 D4 0A B7 D4 0A 81 51 27 7B :D0
5D50 81 5F 10 27 00 82 81 41 10 27 01 C1 81 4F 10 27 :5B
5D60 01 C1 81 10 10 27 02 6C 81 11 10 27 02 84 81 A1 :69
5D70 10 27 02 82 81 A2 10 27 02 9A 81 A3 10 27 02 B2 :C0
5D80 81 31 10 27 03 A0 81 03 10 27 04 D4 81 02 10 27 :D9
5D90 04 6C 81 81 10 27 02 B8 81 90 10 27 03 08 81 9F :D6
5DA0 10 27 03 1D 81 30 10 27 03 43 81 01 10 27 03 AC :ED
5DB0 81 04 10 27 04 E9 81 05 10 27 05 17 81 20 10 26 :59
5DC0 FF 6F B6 D4 01 B7 D3 F9 16 FF 66 31 8D 06 2D B6 :9E
5DD0 D3 FA 10 26 00 AA 20 0B 31 8D 06 40 B6 D3 FA 10 :6F
5DE0 26 00 9D CE D3 81 BE D3 FE F6 D3 FB 26 0B 8D 6B :61
5DF0 30 89 80 00 8D 65 16 FF 38 C1 01 26 12 30 89 05 :30
---------------------------------------------------------
Sum: 3D 1C DB 25 DB E4 74 ED 10 BA 27 50 A6 2F B7 77 :BD


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5E00 00 31 A8 C0 8D 55 30 89 C0 00 8D 4F 16 FF 22 C1 :C8
5E10 02 26 12 30 89 0A 00 31 A8 C0 8D 3F 30 89 C0 00 :DB
5E20 8D 39 16 FF 0C C1 03 26 16 30 89 CF 00 31 A8 80 :C8
5E30 8D 29 31 8D 05 26 30 89 40 00 8D 1F 16 FE F2 C1 :0B
5E40 04 10 26 FE EC 30 89 D4 00 31 A8 80 8D 0D 31 8D :62
5E50 05 0A 30 89 40 00 8D 03 16 FE D6 34 70 C6 0B A6 :9D
5E60 C0 27 16 34 74 C6 08 6F 1F A6 A0 A7 80 EE A1 EF :EC
5E70 84 30 88 4F 5A 26 F0 35 74 30 04 5A 26 E1 35 F0 :5E
5E80 CE D3 81 BE D3 FE F6 D3 FB 26 0B 8D 6B 30 89 80 :D7
5E90 00 8D 65 16 FE 9B C1 01 26 12 30 89 05 00 31 A8 :32
5EA0 C0 8D 55 30 89 C0 00 8D 4F 16 FE 85 C1 02 26 12 :8B
5EB0 30 89 0A 00 31 A8 C0 8D 3F 30 89 C0 00 8D 39 16 :7D
5EC0 FE 6F C1 03 26 16 30 89 CF 00 31 A8 80 8D 29 31 :35
5ED0 8D 04 89 30 89 40 00 8D 1F 16 FE 55 C1 04 10 26 :23
5EE0 FE 4F 30 89 D4 00 31 A8 80 8D 0D 31 8D 04 6D 30 :2C
5EF0 89 40 00 8D 03 16 FE 39 34 70 C6 0B A6 C0 27 16 :BE
---------------------------------------------------------
Sum: 39 A2 B4 D3 32 CF 47 C9 B8 86 16 C5 A4 6D 74 01 :12


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5F00 34 74 C6 08 A6 A0 A7 80 EE A1 EF 81 6F 84 30 88 :8D
5F10 4D 5A 26 F0 35 74 30 04 5A 26 E1 35 F0 31 8D 04 :E2
5F20 5B 20 04 31 8D 04 75 BE D3 FE CE D3 81 F6 D3 FB :2B
5F30 C1 05 26 13 30 89 D1 80 8D 69 30 89 40 00 31 8D :B6
5F40 04 1A 8D 5F 16 FD EA C1 04 26 13 30 89 CC 80 8D :97
5F50 52 30 89 40 00 31 8D 04 03 8D 48 16 FD D3 C1 03 :8F
5F60 26 12 31 A8 40 30 89 07 80 8D 38 30 89 C0 00 8D :5C
5F70 32 16 FD BD C1 02 26 12 31 A8 40 30 89 02 80 8D :DE
5F80 22 30 89 C0 00 8D 1C 16 FD A7 C1 01 10 26 FD A1 :94
5F90 31 A9 00 80 30 89 FD 80 8D 09 30 89 80 00 8D 03 :EF
5FA0 16 FD 8E 34 70 C6 0B A6 C0 27 22 34 74 C6 08 6F :AA
5FB0 80 6F 80 6F 84 30 88 4E 5A 26 F4 C6 08 A6 A0 A7 :97
5FC0 80 EE A1 EF 84 30 88 4F 5A 26 F2 35 74 30 04 5A :32
5FD0 26 D5 35 F0 C6 08 BE D3 FE 33 8D 04 5F A6 C0 A7 :AD
5FE0 84 A7 89 C0 00 A7 89 80 00 30 88 50 5A 26 EE 16 :B0
5FF0 FD 3F C6 10 20 E0 33 8D 04 62 BE D3 FE C6 10 A6 :43
---------------------------------------------------------
Sum: 5B 53 16 D2 3D CC F1 59 60 FE 6D 98 EF 60 76 35 :46


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6000 C0 A7 89 80 00 A7 89 C0 00 6F 84 30 88 50 5A 26 :DB
6010 EE 16 FD 1D 33 8D 04 64 C6 10 BE D3 FE A6 C0 A7 :B8
6020 89 80 00 A7 84 6F 89 C0 00 30 88 50 5A 26 EE 16 :78
6030 FC FF 33 8D 04 66 BE D3 FE C6 10 A6 C0 A7 89 C0 :E0
6040 00 A7 84 6F 89 80 00 30 88 50 5A 26 EE 16 FC E1 :0C
6050 BE D3 FE B6 D3 FA 26 4A 30 1F C6 08 6F 84 6F 89 :8A
6060 C0 00 6F 89 80 00 30 88 50 5A 26 F0 BE D3 FE 33 :72
6070 8D 04 49 C6 08 A6 C0 A7 89 C0 00 A7 89 80 00 A7 :55
6080 80 A6 C0 A7 89 C0 00 A7 89 80 00 A7 80 A6 C0 A7 :BA
6090 89 C0 00 A7 89 80 00 A7 84 30 88 4E 5A 26 D6 16 :96
60A0 FC 8F 30 03 20 B4 C6 08 BE D3 FE 8D 03 16 FC 81 :12
60B0 6F 84 6F 89 C0 00 6F 89 80 00 30 88 50 5A 26 F0 :9B
60C0 39 BE D3 FE C6 08 6F 89 C0 00 6F 89 80 00 6F 80 :B5
60D0 6F 89 C0 00 6F 89 80 00 6F 80 6F 89 C0 00 6F 89 :CF
60E0 80 00 6F 84 30 88 4E 5A 26 DC 16 FC 44 8E 00 00 :B9
60F0 6F 89 40 00 6F 89 80 00 6F 80 8C 3F FF 26 F1 8E :0E
---------------------------------------------------------
Sum: 49 03 94 A1 65 BF DC 22 64 5D 56 15 F4 A0 81 AC :90

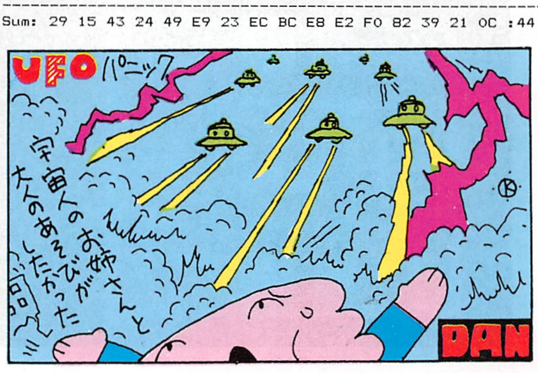

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6100 B4 77 C6 04 31 8D 03 D4 30 88 10 34 34 C6 10 86 :16
6110 03 EE A1 EF 81 4A 26 F9 30 88 4A 5A 26 F1 35 34 :47
6120 5A 26 E5 16 FC 0B 33 8D 02 12 C6 08 BE D3 FE A6 :59
6130 C0 A7 89 C0 00 6F 89 80 00 6F 80 A6 C0 A7 89 C0 :6D
6140 00 6F 89 80 00 6F 80 A6 C0 A7 89 C0 00 6F 89 80 :35
6150 00 6F 84 30 88 4E 5A 26 D6 16 FB D5 86 05 B0 D3 :43
6160 FC B7 D3 FC 49 BB D3 FC B7 D3 FC 80 03 B7 D3 FA :E2
6170 7F D3 FB BE D3 FE C6 08 B6 D3 FD 26 1D 30 89 FD :29
6180 80 17 FF 2C BE D3 FE 8D 32 30 88 50 7C D3 FB B6 :18
6190 D3 FC B1 D3 FB 26 F0 16 FB 97 30 89 02 80 17 FF :5D
61A0 0F BE D3 FE 30 89 02 30 8D 11 30 88 B0 7C D3 FB :D9
61B0 B6 D3 FC B1 D3 FB 26 F0 16 FB 76 8D 27 8A F0 F6 :C5
61C0 D3 FB F1 D3 FA 25 04 A4 1F A7 1F 8D 17 84 E7 A4 :F1
61D0 84 A7 84 8D 0F 8A 0F F6 D3 FB F1 D3 FA 25 04 A4 :33
61E0 01 A7 01 39 B6 D3 F0 F6 D3 F0 BB D3 F0 B7 D3 F0 :0C
61F0 BB D3 F0 F7 D3 F0 BB D3 F0 4C B7 D3 F0 39 86 06 :41
---------------------------------------------------------
Sum: 77 5A 95 71 A0 B6 2C D0 EA A5 FD 6B C4 7E 7A 4E :2A
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6200 B7 D3 FC BE D3 FE 30 89 C0 00 33 8D 03 2E C6 08 :4D
6210 8D 1B 8D 44 BE D3 FE 30 89 C0 00 33 8D 03 3D C6 :47
6220 08 8D 0A 8D 33 7A D3 FC 26 D9 16 FB 04 A6 C0 6F :91
6230 89 C0 00 6F 89 40 00 A7 80 A6 C0 6F 89 C0 00 6F :35
6240 89 40 00 A7 80 A6 C0 6F 89 C0 00 6F 89 40 00 A7 :ED
6250 84 30 88 4E 5A 26 D6 39 8E 50 00 30 1F 26 FC 39 :A1
6260 CE D3 81 8E 30 30 AF C1 AF C1 AF C1 6F C4 FC D3 :62
6270 FC CE D3 81 8E 27 10 8D 16 8E 03 E8 8D 11 8E 00 :2B
6280 64 8D 0C 8E 00 0A 8D 07 C3 00 30 E7 C4 20 10 BF :B6
6290 D3 FA 6C C4 B3 D3 FA 24 F9 6A C0 F3 D3 FA 39 BE :7B
62A0 D3 FE CE D3 81 8D 07 A6 C4 26 FA 16 FA 83 A6 C0 :0A
62B0 C6 08 3D C3 D8 00 1F 02 C6 08 34 10 A6 A0 A7 84 :4A
62C0 A7 89 C0 00 A7 89 80 00 30 88 50 5A 26 EE 35 10 :5B
62D0 30 01 39 BE D3 FE 30 89 C0 00 B6 D3 FA 26 30 B6 :01
62E0 FC FB 31 8D 02 96 C6 08 6F 1F 6F 89 BF FF EE A1 :EE
62F0 EF 89 C0 00 EF 81 EE A1 EF 89 C0 00 EF 81 EE A1 :6E
---------------------------------------------------------
Sum: 3E E7 DC 35 5C B6 67 57 5F 66 0E 28 C6 A3 20 28 :B2


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6300 EF 89 C0 00 EF 84 30 88 4C 5A 26 DC 16 FA 22 31 :6E
6310 8D 02 69 C6 08 EE A1 EF 89 C0 00 EF 81 EE A1 EF :7B
6320 89 C0 00 EF 81 EE A1 EF 99 C0 00 EF 81 6F 89 C0 :A8
6330 00 6F 84 30 88 4A 5A 26 DC 16 F9 F5 00 C3 00 30 :48
6340 3C 0C 0C 00 30 03 00 C0 F0 00 0F 03 00 C0 0C 3C :51
6350 30 30 C3 0C 00 00 00 00 00 00 00 00 00 00 00 00 :2F
6360 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6370 00 00 00 00 00 00 00 00 00 00 00 00 00 FF 00 3F :3E
6380 FF FC FF FF FF FC 3C 3F FF FF FF 03 C3 C0 0F 3C :3D
6390 F0 F0 00 0F 00 00 00 00 00 00 00 00 00 FF 00 3F :2D
63A0 FF FC FF FF FF FC 3C 3F FF FF FF 0F C3 F0 3C 3C :A6
63B0 3C 0F 00 F0 00 00 00 00 00 00 00 00 03 00 30 30 :9E
63C0 C0 C3 33 FF F3 33 3F 33 0F FF FC 03 FF F0 03 00 :4C
63D0 30 0C 00 0C 00 00 00 00 00 00 00 00 03 00 30 30 :AB
63E0 C3 C3 03 FF F0 0F 3F 3C 3F FF FF 33 FF F3 33 00 :97
63F0 33 30 F3 C3 00 00 00 00 00 00 00 00 00 3C 00 00 :55
---------------------------------------------------------
Sum: 81 AF A3 BB 11 E7 C2 39 76 EC 27 FA A2 A7 39 A2 :28


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6400 FF 00 03 FF C0 0F 3C F0 0F FF F0 00 C3 00 03 3C :FC
6410 C0 0C C3 30 00 00 00 00 00 00 00 00 3C 00 00 :FB
6420 FF 00 03 FF C0 0F 3C F0 0F FF F0 03 3C C0 0C 00 :05
6430 30 03 00 C0 00 00 00 00 00 00 00 00 00 0C 0C :0B
6440 0C 0C 0C 00 00 00 00 00 00 00 00 00 00 00 00 00 :24
6450 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6460 00 00 00 00 00 08 08 49 49 08 08 00 00 00 00 00 :B2
6470 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6480 00 00 00 00 00 33 3F 0C 0C 0C 0C 00 00 00 00 00 :A2
6490 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
64A0 00 00 00 00 30 0C 03 0C 30 0C 03 00 00 00 00 00 :8A
64B0 00 00 00 00 00 00 00 00 00 00 00 00 00 0C 00 00 :0C
64C0 3F 00 00 3F 00 0F FF FC 3F FF FF 3F FF FF 3F FF :40
64D0 FF 3F FF FF 00 00 00 00 00 00 00 00 00 FF FF FF :39
64E0 FF 00 03 FF FF FF FF C0 0F FF FF FF FF F0 3F FF :F7
64F0 FF FF FF FC FF FF FF FF FF FF FF FF FF FF FF FF :ED
---------------------------------------------------------
Sum: 36 59 D6 27 AE 72 BF FC F0 1B F4 40 FC F5 97 44 :72


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6500 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
6510 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
6520 FF FF FF FF FF FF 00 00 FF FF FF FF 00 00 FF FF :F4
6530 FF FC 00 00 3F FF FF F0 00 00 0F FF 00 00 00 00 :36
6540 00 00 03 03 00 C0 C0 0C 00 0C C0 33 03 0C C0 CC :2C
6550 33 03 3F CC 00 00 00 00 00 00 00 00 C0 30 C0 0C :FD
6560 CC 0C 03 03 00 30 C0 33 C0 F0 CC 0C C3 00 C3 0C :1B
6570 30 0F 3F C3 00 00 00 00 00 00 00 00 00 00 0F F0 :40
6580 00 00 00 0F FF FF F0 00 00 FF FF FF FF 00 03 FF :FB
6590 FF FF FF C0 0F FF F3 CF FF F0 3C F3 FF FF CF 3C :B4
65A0 FF FF C3 C3 FF FF C3 F0 00 00 0F C3 00 00 00 00 :07
65B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
65C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
65D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
65E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
65F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
---------------------------------------------------------
Sum: 29 15 43 24 49 E9 23 EC BC E8 E2 F0 82 39 21 0C :44
```



FM-8

Kコンパイラ・ファイル版 パックマン

■COMPAC S

“高級言語で高速ゲームを作る” 例として，Kコンパイラでパックマンを作ってみました．フル・カラー，高速（本物よりはちょっと遅い？）のパックマンを楽しんでください．

1．ゲームについて

有名なゲームなので簡単に説明します．パックマンがエサをパクパクと食べ，4匹（頭？）のモンスターがそれをじゃまをします．4匹のモンスターはそれぞれ性格が違っていて，しつこく追いかけるもの，待ちぶせするもの，気まぐれで追いかけたり待ちぶせするもの，まったく気まぐれのものとあります．それぞれアカベー，ピンキー，アオスケ，グズタと愛称がつけられています．

モンスターは真中の部屋から次々と出て来てパックマンを食べようとします．パックマンは，テン・キーの2，4，6，8により左右上下に動き，モンスターを避けながら，エサを食べます．

パックマンがパワーエサを食べると，一定の時間だけモンスターを食べることができます．モンスターは連続して食べていくと得点が，200，400，800，1000と増えます．

ときどき中央の通路にフルーツが表示されます．これを食べるとボーナス得点がもらえます．エサは食べると10点，パワーエサは50点になります．

パックマンは3人いて，モンスターに食べられると死にます．得点が10,000点を越えるとパックマンが1人増えます．

ゲームのスタート番地は$5000です．

2．プログラムの作り方

BASICのコメント文（'）の形式で**リスト1**を全部打込みます．それをファイルにASCIIセーブします．

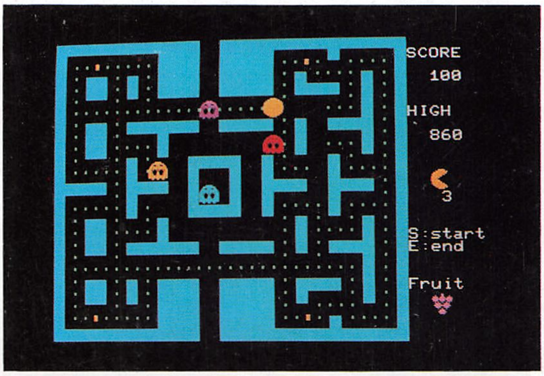
ファイル版Kコンパイラでコンパイルします．コンパイル・エラーがなくなるようデバッグしてください．

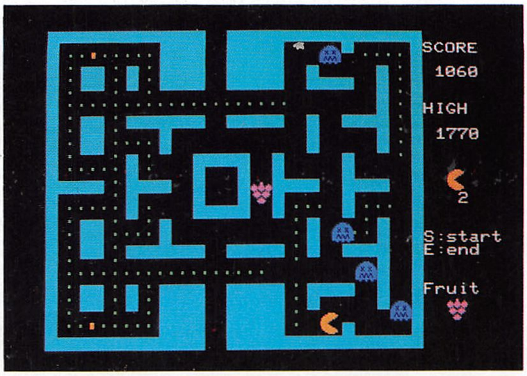
次に，MONコマンドでモニタ・モードにして，Mコマンドで**リスト2**のデータを打ち込みます．必ずチェック・サムを確認しておいてください．

全部できたら，SAVEMコマンドで，$4D00から$6F5Fまでを，スタート番地$5000にしてSAVEします．これで，LOADM “ファイル名”,,R RETURN で実行します．

3．プログラムの説明

プログラムはすべてKコンパイラで書かれています．Kコンパイラの引数付ブルーチン／関数コールはなかなかか強力なので，各処理は極力サブルーチン／関数化してメモリを節約しています．ラベルはそれぞれ意味を持たせてい





るので，解読のときコメント代わりになります．

グラフィックおよびコンソール制御はサブシステムにコマンドを送って実行させています．キー入力ルーチンだけはサブシステム内のアドレスを直接参照しています．これはサブシステムのコマンドを使うと，キー入力時に一定の待ち時間ができ，ゲームとしては操作性がよくないためです．このため，サブシステム・テスト・コマンドを使っています．

◘参考文献

1) ルリタテハ：“PUCKMAN”，I/O，'81年12月号

表1．データ・エリア(先頭＄6A80)

アドレス	内容
0	迷路データ
300	パックマン
332	〃
364	〃
396	〃
3C8	〃
3FA	〃
42C	モンスター
45E	いじけモンスター
490	フルーツ
4C2	エサ
4C4	パワーエサ
4C8	音1
4CA	音2
4CD〜4D1	音3

リスト1

```
100 'CONST DATA=$6A80;WORK=$4000;CODE $1A,$40;(* INHIBIT INTERRUPTION *);CODE $1
0,$CE,$3F,$FF;(* SET STACK POINTER *)
110 'GA=DATA;P=GA+768;M=P+50*6;F=M+50*2;DT=F+50;PWD=DT+2;S1=PWD+4;S2=S1+2;S3=S2+
3
120 'GB=WORK;R=GB+768;MY=R+8;MX=MY+8;DD=MX+8;U=DD+8;V=U+8;MC=V+8;KC=MC+8;PP=KC+8
;WKA=PP+8;WKEND=WKA+64
130 'PP(0)=P+50;FOR I=1 TO 3;PP(I)=PP(I-1)+50;NEXT;M1=M+50
140 'KC(0)=2;KC(1)=3;KC(2)=6;KC(3)=5
150 'T=16;G=8;SP=3;D=1;PW=2;FR=4
160 'HSC=0;WS=162
170 'CON[];LINE[0,0,33*T-1,25*G-1,0,5]
180 'FOR I=0 TO 31;FOR J=0 TO 23;Q=GA:I+J*32);IF Q THEN BOX1[I,J];FI;NEXT;NEXT
190 'CL=7;LOC[33,1];?"SCORE";LOC[33,6];?"HIGH";PX=35;PY=11;DP[PP(1),0];LOC[33,16
];?"S:start";LOC[33,17];?"E:end";LOC[33,20];?"Fruit";FRT[35,21,0]
200 'INIT;SC=0;PK=3;FU=-10;PF=0
210 'CL=7;LOC[33,3];?#5,SC;LOC[36,13];?PK
220 'INIT1;NC=218;W=-1;FT=-1;FRT[18,12,1];DFR=-1;KFR=-1
230 'FOR J=0 TO 23;FOR I=0 TO 31;K=I+J*32;B=GA:K);GB:K)=B;IF B=1 THEN DOT[I,J,0]
;ELSE IF B=2 THEN POW[I,J,0];FI;FI;;NEXT;NEXT
240 'WS=WS-2;WS4=WS/4;FU=FU+20
250 'INIT2;FOR I=0 TO 3;MC(I)=(I+1)*5;NEXT
260 'PX=24;PY=12;FOR I=0 TO 3;MX(I)=14;MY(I)=12;U(I)=0;V(I)=0;DD(I)=0;NEXT;CK=0;
PEAT=0;SCC=0
270 'LOOP;K=CK-CK/4*4;IF MC(K)>0 THEN MDANC[];ELSE MMOVE[];FI
280 'IF PEAT THEN GOTO EATED;FI;PMOVE[]
290 'IF SCC<>0 THEN SC=SC+SCC;SCC=0;CL=7;LOC[33,3];?#4,SC,"0";IF SC>1000 AND PF=
0 THEN PK=PK+1;PF=1;CL=6;LOC[36,13];?PK;FOR I=0 TO 10;SND[S3];NEXT;FI;FI
300 'IF NC<=0 THEN DP[P,1];FOR K=0 TO 3;BOX[MX(K),MY(K)];NEXT;GOTO INIT1;FI
310 'IF FT>0 THEN FT=FT-1;ELSE IF FT=0 THEN FRT[18,12,1];SET[18,12,SP];FT=-1;FI;
FI
320 'IF W>0 THEN W=W-1;ELSE IF W=0 THEN FOR K=0 TO 3;IF DD(K)>0 THEN DM[M1,1];DD
(K)=0;FI;NEXT;W=-1;FI;FI
330 'IF PEAT=0 THEN CK=CK+1;GOTO LOOP;FI
340 'EATED;DP[P,0];WAIT[1];FOR I=0 TO 3;DP[PP(I),2];SND[S3];NEXT;DP[P,1];SND[S3]
;P4=PP(3)+50;DP[P4,0];WAIT[2];SND[S1];DP[P4,1];FOR K=0 TO 3;X=MX(K);Y=MY(K);BOX[
X,Y];DD(K)=0;WPLT[X,Y];NEXT
350 ' PK=PK-1;CL=7;LOC[36,13];?PK;IF PK=0 THEN IF SC>HSC;HSC=SC;CL=7;LOC[33,8];?
#4,HSC,"0";FI;LOC[12,16];?"GAME OVER"
360 'START;A=KEY[];IF A='S' OR A='s' THEN LOC[12,16];?"             ";GOTO INIT;FI;I
F A='E' OR A='e' THEN CODE $1C,$40;END FI;GOTO START;ELSE GOTO INIT2;FI
370 'PMOVE;CKK=CK-CK/4*4;IF CKK=0 THEN DP[P,0];RETURN;FI;IF CKK=2 THEN GOTO PMOV
E1;ELSE RETURN;FI
380 'PMOVE1;N=KEY[]-$30
390 'DX=0;DY=0;IF N=6 THEN PI=1;DX=1;ELSE IF N=8 THEN PI=0;DY=-1;ELSE IF N=4 THE
N PI=2;DX=-1;ELSE IF N=2 THEN PI=3;DY=1;ELSE RETURN;FI;FI;FI;FI
400 'DP[P,1]
410 'IF DY=-1 THEN IF PY=0 THEN PY=24;FI;ELSE IF DY=1 THEN IF PY=23 THEN PY=0;FI
;FI;FI
420 'J=SCR[PX+DX,PY+DY]
430 'IF J=0 THEN DP[PP(PI),0];RETURN;FI
440 'PX=PX+DX;PY=PY+DY;DP[PP(PI),0];SET[PX,PY,SP]
450 'FOR K=0 TO 3;IF PX=MX(K) AND PY=MY(K) THEN GOTO PENT1;FI;NEXT
460 'IF J=D THEN SCC=1;SND[S1];GOTO PEND;FI
470 'IF J=PW THEN SCC=5;SI=20;SND[S2];W=WS;CC=1;FOR K=0 TO 3;IF MC(K)=0 THEN BOX
[MX(K),MY(K)];DD(K)=1;DM[M1,0];U(K)=-U(K);V(K)=-V(K);FI;NEXT;GOTO PEND;FI
480 'IF J=FR THEN SND[S3];NDS[18,12,FU];RETURN;FI
490 'RETURN
```

パックマン

リスト1

```
500 'PENT1;IF J=D OR J=PW THEN PEND[];FI;IF DD(K)=0 THEN PEAT=1;RETURN;FI
510 'PENT;SND[S2];NDS[PX,PY,SI];MX(K)=14;MY(K)=11;DD(K)=0;MC(K)=15;SI=SI*2;RETUR
N
520 'PEND;NC=NC-1;IF NC=70 OR NC=140 THEN FRT[18,12,0];SET[18,12,FR];FT=WS;FI;RE
TURN
530 'MDANC;CC=KC(K);IF MC(K)>1 THEN MC(K)=MC(K)-1;DM[M,1];MX(K)=14+RND(2);MY(K)=
11+RND(2);DM[M,0];RETURN
540 'ELSE MC(K)=0;DM[M,1];MX(K)=11;MY(K)=11;U(K)=0;V(K)=-1;DM[M,0];RETURN;FI
550 'MMOVE;IF DD(K)>0 THEN IF CK-CK/16*16=K+12 THEN WAIT[1];RETURN;FI;FI
560 'DW=DD(K);DX=U(K);DY=V(K)
570 'MI=M;IF DW>0 THEN MI=M1;FI
580 'DM[MI,1];X=MX(K);Y=MY(K);WPLT[X,Y];IF K=KFR THEN IF DFR>0 THEN DFR=0;ELSE I
F DFR=0 THEN FRT[18,12,0];DFR=-1;KFR=-1;FI;FI;FI
590 'HK=(SCR[X-1,Y]=0)+(SCR[X+1,Y]=0)+(SCR[X,Y-1]=0)+(SCR[X,Y+1]=0)
600 'IF HK>1 THEN GOTO HOKO;FI;IF K=0 THEN AKA[];ELSE IF K=1 THEN PIN[];ELSE IF
K=2 THEN GUZ[];ELSE AWO[];FI;FI;FI
610 'HOKO;X=MX(K)+DX;Y=MY(K)+DY
620 'IF DY=-1 THEN IF Y=-1 THEN Y=23;ELSE IF DY=1 THEN IF Y=24 THEN Y=0;FI;FI;FI
;FI
630 'J=SCR[X,Y]
640 'IF J=0 THEN IF DY=0 THEN DX=0;DY=SGN(PY-MY(K))*(1-DW*2);IF DY=0 THEN DY=1;F
I;IF SCR[MX(K),MY(K)+DY]=0 THEN DY=-DY;FI;ELSE DY=0;DX=SGN(PX-MX(K))*(1-DW*2);IF
 DX=0 THEN DX=1;FI;IF SCR[MX(K)+DX,MY(K)]=0 THEN DX=-DX;FI;FIGOTO HOKO;FI
650 'IF J=FR THEN DFR=1;KFR=K;FI
660 'IF X=PX AND Y=PY THEN IF DW>0 THEN GOTO PENT;ELSE PEAT=1;FI;FI
670 'CC=KC(K);IF DW THEN IF W>WS4 THEN CC=1;ELSE IF W>0 THEN CC=7;FI;FI;FI
680 'MX(K)=X;MY(K)=Y;DM[MI,0];U(K)=DX;V(K)=DY;RETURN
690 'AKA;DX=SGN(PX-X);DY=0;IF DX=0 THEN DY=SGN(PY-Y);FI;GOTO MRET
700 'PIN;DY=SGN(PY-Y);DX=0;IF DY=-V(K) THEN DY=V(K);GOTO MRET;FI;IF DY=0 THEN DX
=SGN(PX-X);FI;MRET;IF DW>0 THEN DX=-DX;DY=-DY;FI;RETURN
710 'GUZ;IF RND(2) THEN DX=RND(2)*2-1;DY=0;ELSE DX=0;DY=RND(2)*2-1;FI;RETURN
720 'AWO;Q=RND(3);IF Q=0 THEN GOTO AKA;FI;IF Q=1 THEN GOTO PIN;FI;GOTO GUZ
730 'WAIT;FOR Q=0 TO %1;FOR QQ=0 TO 1500;NEXT;NEXT;RETURN
740 'SCR;DMY=GB:%1+%2*32);RETURN
750 'SET;GB:%1+%2*32)=%3;RETURN
760 'WPLT;B=SCR[%1,%2];IF B=D THEN DOT[%1,%2,0];ELSE IF B=PW THEN POW[%1,%2,0];E
LSE IF B=FR THEN FRT[%1,%2,0];FI;FI;FI;RETURN
770 'FRT;BOX[%1,%2];XX=%1*T+T;YY=%2*G+G;PUT[XX-14,YY-6,XX+14,YY+6,F,50,3,%3];RET
URN
780 'DP;XX=PX*T+T;YY=PY*G+G;PUT[XX-14,YY-6,XX+14,YY+6,%1,50,6,%2];RETURN
790 'DM;XX=MX(K)*T+T;YY=MY(K)*G+G;PUT[XX-14,YY-5,XX+14,YY+6,%1,50,CC,%2];RETURN
800 'POW;XX=%1*T+T;YY=%2*G+G;PUT[XX-2,YY-1,XX+2,YY+2,PWD,3,6,%3];RETURN
810 'DOT;XX=%1*T+T;YY=%2*G+G;PUT[XX-1,YY,XX+1,YY+1,DT,2,4,%3];RETURN
820 'SND;WK=%1;Q=0;WHILE WK:Q)<>0;SN=WK:Q);FOR Z1=1 TO 5;FOR Z=0 TO SN;POKE $FD0
3,$81;NEXT;FOR Z=0 TO SN;POKE $FD03,$80;NEXT;POKE $FD03,$80;NEXT;Q=Q+1;WEND;RETU
RN
830 'NDS;BOX[%1,%2];XZ=%1*T-T;ZZ=%1-1;YZ=%2*G+G;GET[XZ,YZ,XZ+63,YZ+7,WKA,64,5]
840 'CL=7;LOC[ZZ,%2+1];?#3,%3,"0";SCC=%3;FOR O=0 TO 5;SND[S2];NEXT;LOC[ZZ,%2+1];
?"     ";PUT[XZ,YZ,XZ+63,YZ+7,WKA,64,5,0];FOR O=ZZ TO ZZ+3;WPLT[O,%2];NEXT;RETURN
850 'BOX1;XX=%1*T;YY=%2*G;LINE[XX,YY,XX+T+T-1,YY+G+G-1,0,0];RETURN
860 'BOX;XX=%1*T+2;YY=%2*G+2;LINE[XX,YY,XX+T+T-3,YY+G+G-3,0,0];RETURN
870 '(* GRAPHIC SUBROUTINES *)
880 'PUT;QE=%6-1;HLT[];ADDR=$FC82;CPOKE ADDR,$1C,%1#,%2#,%3#,%4#,%7,%8,%6;DTD=%5
;FOR WKI=0 TO QE;ADDR:WKI)=DTD:WKI);NEXT;SUB[];RETURN
890 'GET;QE=%6-1;HLT[];ADDR=$FC82;CPOKE ADDR,$1B,%1#,%2#,%3#,%4#,1,%7;SUB[];CODE
 $B6,$18,$4A,$26,$FD;HLT[];DTD=%5;ADDR=$FC84;FOR WKI=0 TO QE;DTD:WKI)=ADDR:WKI);
NEXT;CODE $F6,$FC80#,$CA80#,$F7,$FC80#;SUB[];RETURN
900 'LINE;HLT[];ADDR=$FC82;CPOKE ADDR,$15,%6,%5,%1#,%2#,%3#,%4#;SUB[];RETURN
910 'LOC;HLT[];ADDR=$FC82;CPOKE ADDR,3,5,$12,%1,%2,$11,CL+$80;SUB[];RETURN
920 'CON;HLT[];ADDR=$FC82;CPOKE ADDR,1,0,40,25,0,25,0,1,0;SUB[];RETURN
930 'KEY;HLT[];ADDR=$FC82;CPOKE ADDR,$3F,'Y','A','M','A','U','C','H','I',$93,$D3
8F#,$90,$B6,$D401#,$B7,$D382#,$B6,$D40A#,$39;SUB[];HLT[];KY=PEEK($FC82);SUB[];DM
Y=KY;RETURN
940 'HLT;CODE $B6,$FD,$05,$2B,$FB;POKE $FD05,$80;CODE $B6,$FD,$05,$2A,$FB;RETURN
950 'SUB;POKE $FD05,0;RETURN
```

リスト2

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6A80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 :03
6A90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6AA0 00 01 01 02 01 01 01 01 01 00 00 00 00 00 03 00 :0C
6AB0 00 00 00 00 00 01 01 02 01 00 00 01 01 01 01 00 :09
6AC0 00 01 00 00 00 01 00 00 01 00 00 00 00 00 03 00 :06
6AD0 00 00 00 00 00 01 00 00 01 01 01 01 00 00 01 00 :06
6AE0 00 01 00 00 00 01 00 00 01 00 00 00 00 00 03 00 :06
6AF0 00 00 00 00 00 01 00 00 00 00 00 01 00 00 01 00 :03
---------------------------------------------------------
Sum: 00 03 01 02 01 06 02 03 05 01 01 03 01 01 0F 00 :2D


Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
```

```
6B00 00 01 00 00 00 01 00 00 01 00 00 00 00 00 03 00 :06
6B10 00 00 00 00 00 01 00 00 00 00 00 01 00 00 01 00 :03
6B20 00 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 :0F
6B30 01 01 01 01 01 01 01 01 01 01 01 01 00 00 01 00 :0D
6B40 00 01 00 00 00 01 00 00 00 00 00 00 00 00 03 00 :05
6B50 00 00 00 00 00 01 00 00 01 00 00 00 00 00 01 00 :03
6B60 00 01 00 00 00 01 00 00 00 00 00 00 00 00 03 00 :05
6B70 00 00 00 00 00 01 00 00 01 00 00 00 00 00 01 00 :03
6B80 00 01 00 00 00 01 01 01 01 00 00 03 03 03 03 03 :14
6B90 03 03 03 03 03 01 00 00 01 01 01 01 00 00 01 00 :15
6BA0 00 01 00 00 00 01 00 00 01 00 00 03 00 00 00 00 :06
6BB0 00 00 03 00 00 01 00 00 01 00 00 01 00 00 01 00 :07
6BC0 00 01 01 01 01 01 00 00 01 03 03 03 00 00 00 00 :0F
6BD0 00 00 03 00 00 01 01 01 01 00 00 01 01 01 01 00 :0B
```

リスト 2

```
6BE0 00 00 00 00 00 01 00 00 00 00 00 03 00 00 03 03 :0A
6BF0 00 00 03 00 00 00 00 00 03 00 00 00 00 00 01 00 :07
-------------------------------------------------------
Sum: 04 0B 0F 06 06 0F 04 04 0E 06 06 12 05 05 18 07 :96


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6C00 00 00 00 00 00 01 00 00 00 00 00 03 00 00 03 03 :0A
6C10 00 00 03 00 00 00 00 00 03 00 00 00 00 00 01 00 :07
6C20 00 01 01 01 01 01 00 00 01 03 03 03 00 00 00 00 :0F
6C30 00 00 03 00 00 01 01 01 01 00 00 01 01 01 01 00 :0B
6C40 00 01 00 00 00 01 00 00 01 00 00 03 00 00 00 00 :06
6C50 00 00 03 00 00 01 00 00 01 00 00 01 00 00 01 00 :07
6C60 00 01 00 00 00 01 01 01 01 00 00 03 03 03 03 03 :14
6C70 03 03 03 03 03 01 00 00 01 01 01 01 00 00 01 00 :15
6C80 00 01 00 00 00 01 00 00 00 00 00 00 00 00 03 00 :05
6C90 00 00 00 00 00 01 00 00 01 00 00 00 00 00 01 00 :03
6CA0 00 01 00 00 00 01 00 00 00 00 00 00 00 00 03 00 :05
6CB0 00 00 00 00 00 01 00 00 01 00 00 00 00 00 01 00 :03
6CC0 00 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 :0F
6CD0 01 01 01 01 01 01 01 01 01 01 01 01 00 00 01 00 :0D
6CE0 00 01 00 00 00 01 00 00 01 00 00 00 00 00 03 00 :06
6CF0 00 00 00 00 00 01 00 00 00 00 00 01 00 00 01 00 :03
-------------------------------------------------------
Sum: 04 0B 0F 06 06 0F 04 04 0E 06 06 12 05 05 18 07 :96


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6D00 00 01 00 00 00 01 00 00 01 00 00 00 00 00 03 00 :06
6D10 00 00 00 00 00 01 00 00 00 00 00 01 00 00 01 00 :03
6D20 00 01 00 00 00 01 00 00 01 00 00 00 00 00 03 00 :06
6D30 00 00 00 00 00 01 00 00 01 01 01 01 00 00 01 00 :06
6D40 00 01 01 02 01 01 01 01 01 00 00 00 00 00 03 00 :0C
6D50 00 00 00 00 00 01 01 02 01 00 00 01 01 01 01 00 :09
6D60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 :03
6D70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6D80 00 3F E0 00 1F FF F0 07 FF FF F0 FF FF FF E7 FF :05
6D90 FF FF 7F FF FF FF FF FF FF FF FF FF FF 7F FF FF :F0
6DA0 F3 FF FF FF 87 FF FF F0 07 FF FC 00 03 FE 00 3F :A7
6DB0 00 00 00 00 00 00 00 00 00 04 00 00 10 F8 00 03 :0F
6DC0 E7 E0 00 3F 7F C0 07 FF FF 00 7F FF FE 0F FF 7F :53
6DD0 F8 FF F3 FF FF FF 87 FF FF F0 07 FF FC 00 03 FE :5F
```

```
6DE0 00 3F 00 00 00 3F E0 00 1F FF F0 07 FF FF 00 FF :70
6DF0 FF C0 07 FF E0 00 7F F8 00 03 FE 00 00 1F FE 00 :3A
-------------------------------------------------------
Sum: D0 1E 59 3D 04 01 DD EF 27 F4 60 06 0B A2 F5 BC :34


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6E00 00 7F FE 00 03 FF FF 00 07 FF FF 00 07 FF FC 00 :85
6E10 03 FE 00 3F 00 00 00 3F E0 00 1F FF F0 00 7F FF :EB
6E20 F0 00 7F FF E0 00 3F FF 00 00 3F FC 00 00 3F E0 :E6
6E30 00 0F FF 00 03 FF F0 01 FF FF 80 7F FF F0 07 FF :F3
6E40 FC 00 03 FE 00 3F 00 00 00 3F E0 00 1F FF F0 07 :70
6E50 FF FF F0 FF FF FF E7 FF 8F FF 7F F8 3F FF FF 00 :13
6E60 7F FF F0 01 FF 7E 00 03 F3 E0 00 0F 84 00 00 10 :65
6E70 00 00 00 00 00 00 00 3F 00 00 00 00 00 00 00 20 :5F
6E80 00 00 01 00 00 1C 08 0E 00 38 41 C0 00 72 38 00 :16
6E90 00 00 00 0F F8 03 FE 00 00 00 00 07 13 80 00 E0 :82
6EA0 87 00 1C 04 0E 00 00 20 00 3F 00 00 00 3F E0 00 :33
6EB0 1F FF F0 03 83 E0 E0 7B EE FB C7 BF AF EF 79 04 :59
6EC0 41 3F E0 08 03 FF 80 E0 3F FE 0F 83 FF FF FF FF :95
6ED0 FF FF FF FE 0E 0E 0E 00 20 20 20 1F 00 00 00 3F :E3
6EE0 E0 00 1F FF F0 03 FF FF E0 7C 9F 27 C7 F1 FC 7F :44
6EF0 7F 27 C9 FF FF FF FF FF C7 1C 7F FC 92 49 FF CE :70
-------------------------------------------------------
Sum: B2 EE 33 56 6D C8 87 07 5C 44 91 CC F2 46 3B 84 :E0


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6F00 38 E7 FF FF FF FE 0E 0E 0E 00 20 20 20 1F 00 00 :C3
6F10 00 07 00 00 40 38 04 1F C7 F1 FD FF FF FF F7 F1 :3C
6F20 FC 7F 04 41 04 40 1F C1 FC 03 FF BF F8 07 F0 7F :0F
6F30 00 04 10 40 00 07 F0 00 00 FF E0 00 01 FC 00 3F :66
6F40 00 00 FF C0 FF FF F0 00 10 00 80 20 00 80 70 60 :AD
6F50 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :40
-------------------------------------------------------
Sum: 74 71 12 40 42 7C 11 EE E1 F3 7C FE 18 A1 57 0F :61
```





MZ-80Bの移植版

グラフィック花札 こいこい

■COMPAC O.H

MZ-80Bの『花札』をFM-8に移植し，同時にドット単位に色もつけました．640×200の分解能と1ドット単位で色を付けられるFM-8ならではのプログラムです．

一. 長 所

■MZB同様グラフィックをフルに生かしてあります．花札を忠実に再現するため，一枚の札を64×47ドットの大きさにしています．

■他はMZBと同じです．

二. 準 備

❶ 花札の図柄データをマシン語データ領域に書き込むため，CLEAR，&H34F2 CR を実行してください．

❷ MONモードで$34F3番地から書き込んで行きます．リストは$34F0～$34F2がありますが，無視して結構です．データは$7FFF番地までであります．

❸ データの入力が終ったら，STOPまたは，CTRL-C でBASICに戻り，SAVEM “DATA”，&H34F3，&H7FFF，&H34F3 CR で テープにSAVEしてください．

❹ BASICを入力し，SAVEしてできあがりです．

❺ プログラムをLOADするときは必ずCLEAR，&H34F2 CR を入力してください．

三. こいこいの説明

こいこいは2人でやるゲームで，スピード感，スリル，面白さのどれをとっても，馬鹿っ花をしのぐものがあります．

札は48枚すべてを使い，まず，親・子・場に各々8枚ずつ配ります．残りは山札として，場にふせて積重ねておきます．その後は馬鹿っ花（花合わせ）と同様にして，親から手札を1枚場に出し，場札と絵が合えば取れます．次に山札を1枚めくり，場札と合わせます．その手順を親子順番に行なっていくわけです．

馬鹿っ花と違う点は，得点の対象になるのは役のみで，札自体の点数は勝敗には関係がないという点です．つまり，早く役を作った方が勝ちで，役ができた時点で『**勝負**』または『**こいこい**』を宣言します．

『**勝負**』のときはその時点で精算し，役の得点分がもらえます．相手は0文（こいこいの点数単位は文）です．『**勝負**』を宣言すれば，間違いなく勝てるわけですが，応々にして，ひとつの役ができたときは，次々に役ができてしまうものです．だから，見込みのあるときは少々の文数で止めたら損なのです．

役ができた時点で，これからも役ができそうだと思ったら，『**こいこい**』を宣言し，ゲームを続行します．ところが，『**こいこい**』を宣言した場合，もし自分に役ができる前に相手に役ができて『**勝負**』を宣言されれば，自分の負けとなります．ここが，こいこいの面白い要素で，適切な判断を要求されます．

最後まで，両方に役ができない場合は親の勝ちで，子から親に6文渡します．

四. 花札ゲーム・こいこいの遊び方

最下行があなたの手札，右2列が場札です（**写真1**）．親がコンピュータなら，コンピュータから札を合わせ，取札は最上行に並べられます．親があなたなら“**ドレ ヲキリマスカ？**”の質問に数字で入力してやってください．もし，あなたが切った札と同月の札が場に複数枚でていれば（たとえば，あなたが桜の1点札を切り，場に桜の5点札と20点札があった場合），『**何点札と合わせますか？**』と聞いてくるので，5点札なら5，20点札なら20と入力してください．

あとはどんどん役を作っていけばいいのです．コンピュータに役ができれば，『**こいこい**』か『**勝負**』かを自動的に判断して宣言してきます．あなたに役ができれば，『**勝負＝0，こいこい＝1？**』と聞いてくるので0か1を入力してください（**写真2**）．

サービス機能として，コンピュータまたはあなたが『**勝負**』を宣言すれば“野球拳”のメロディが，『**こいこい**』を宣言すれば，“しょじょ寺のたぬき林”のメロディが美しく（？）流れてきます．なぜ，こいこいがしょじょ寺なのか……は，歌詞を口ずさんでもらえばわかります．ねッ？

ゲームは12回で終了となります．くれぐれも．コンピュ

ータのような単純知脳の機械には負けないでください．もし，あなたが負ければ，コンピュータが笑うように作ってあります（これは冗談）．

役は，五光，四光，三光，赤タン，青タン，猪鹿蝶のみ登録してあります．このほかに，月見酒，花見酒，タネ，タン，カスなどがあるのですが，地方によっては役にしないところもあり，また月見酒などはできやすいのであえて採用しませんでした．

写真 1

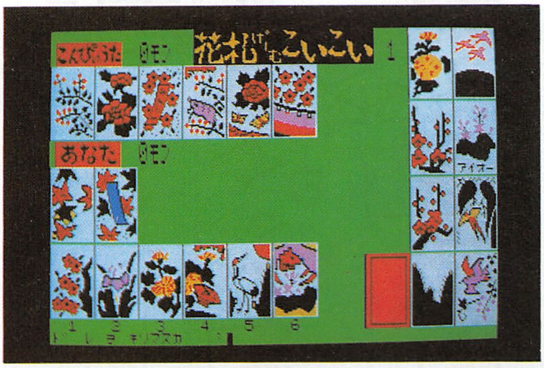
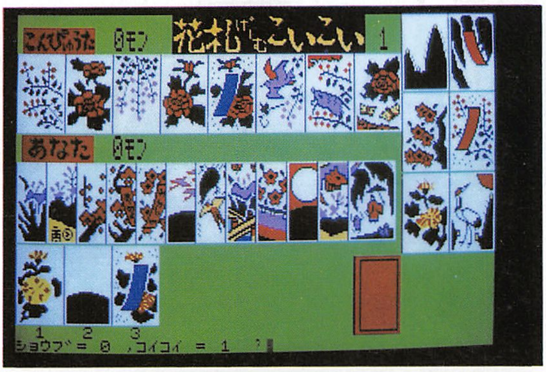


写真 2



五．プログラムの説明

概略のフローチャートを図1に示します．

絵札その他のプログラムは＄34F3から＄7FFFまで入っています．絵札の表示にはPUT＠Aを使っていますが，49枚分の配列変数を持つことはとてもできないので，表示するときに配列Aにデータを読み込ませ PUT＠A で表示させています．また，カラーにするため本来R・G・Bそれぞれのデータを持たせる必要がありますが，これもできるはずがないので，2枚分のデータだけを持たせこれを“AND”，“OR”，“XOR” を上手に使い三色のデータを作り出しています．配列に読み込み三色のデータを作り出すには＄37E6からのマシン語を使っています．

BASICプログラムは820行で親を決めています．変数OYA＝0なら親はコンピュータ，1ならあなたです．

1020行から1300行までで，札を配って（札コードを表2の各変数に割り当てて）います．

1520行から1730行まではコンピュータの札合わせ，1800行から2015行まではあなたの札合わせを行なっています．両者の相違点は，あなたは指定した切り札と場札が合っているかどうかの判定だけを行なっているのに対し，コンピュータは手札と場札のすべての組み合わせの中から最適なものを抽出する必要があるという点です．

グラフィックのデータがこのプログラムの“顔”とするならば，この『最適組み合わせ抽出』は“脳ミソ”にあたる重要な部分と言えます．

改良して，コンピュータをより強くしてみたいと思われる人もおられるでしょうから，この部分のアルゴリズムを記述しておきます．

まず，必要なテーブルが2つあります．ひとつは合致したすべての組み合わせのデータを一時，貯える配列変数CO（i，j）と，あとひとつは役のデータを貯えるTA(i，j）です．変数の内容の意味は表3をみてください．

『最適抽出』の際の優先順位は図2のフローチャートを見てもらえば理解できると思います．5500行から5680行で，CO（i，j）テーブルを作成し，何対の役の組みができたかをCCに入れてリターンします．その結果を5700行から5910行で判断し，最もいい組み合わせを選択してTA(i，j）テーブルに反映しています．

ほぼ，人間が札合わせをするときの思考を再現していますが，大局的な見地にたっての思考はできません．たとえば，手札の中の赤タンの元になるカス札が3枚そろっているから，他のカス札から順に切っていこう…というようなことは考慮できません．

六．最後に……

絵札を見ていただくとわかりますが，さすがFM-8という感じです．スピードはMZBに比べると劣りますが，色の美しさではどのマイコンにも負けないと思います．

表1 役の得点

五光	四光	三光	青丹	赤丹	猪鹿蝶
15文	10文	8文	6文	6文	5文

表2

F(i,j) →i, ↓j

0	1	2	3	4
コンピュータ手札	あなた手札	場札	コンピュータ取札	あなた取札
1～48の札コード				

表3

CO(CC,YC) →CC, ↓YC

YC		0	1	2	3 …
		役札なし	役札あり	→	
0	F(0,j0)のj0の値	0			
1	F(2,j2)のj2の値				
2	TA(XA,Y1)のXAの値	0	0	0	0
3	TA(XB,Y1)のXBの値	0	0	0	0

（注）．プログラム実行前にイニシャル・セット

TA(XA,Y) →XA, ↓Y

Y		0	1 五光	2 四光	3 三光	4 青丹	5 赤丹	6 猪鹿蝶
0	可能性フラグ	9	0	0	0	0	0	0
1	コンピュータ役までの残札数	9	5	4	3	3	3	3
2	あなた役までの残札数	9	5	4	3	3	3	3

可能性フラグ
1…コンピュータが取っているのみ
2…あなたが取っているのみ
3…持ち合い
4…コンピュータ側の役に決定
5…あなた側の役に決定

図１　概略フローチャート

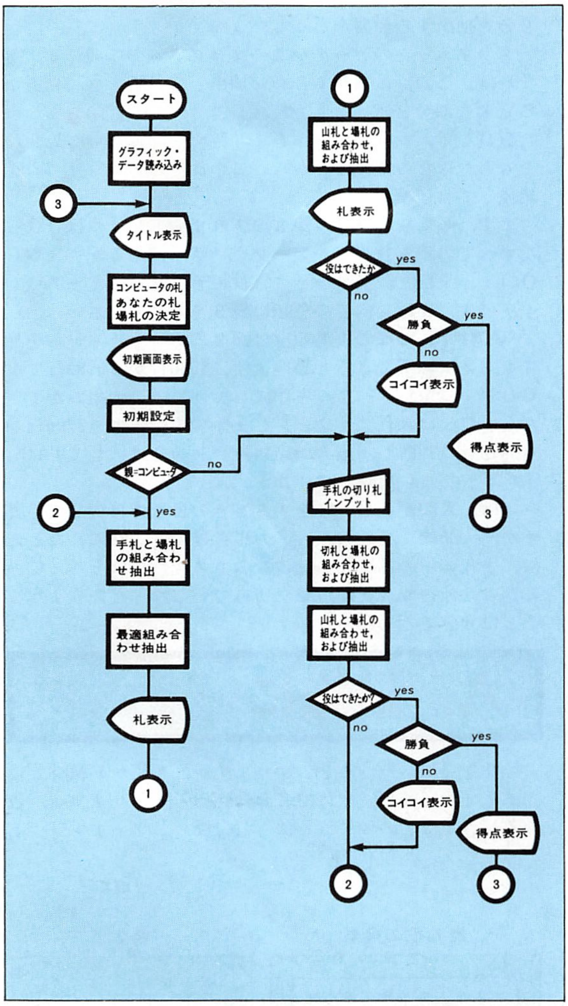
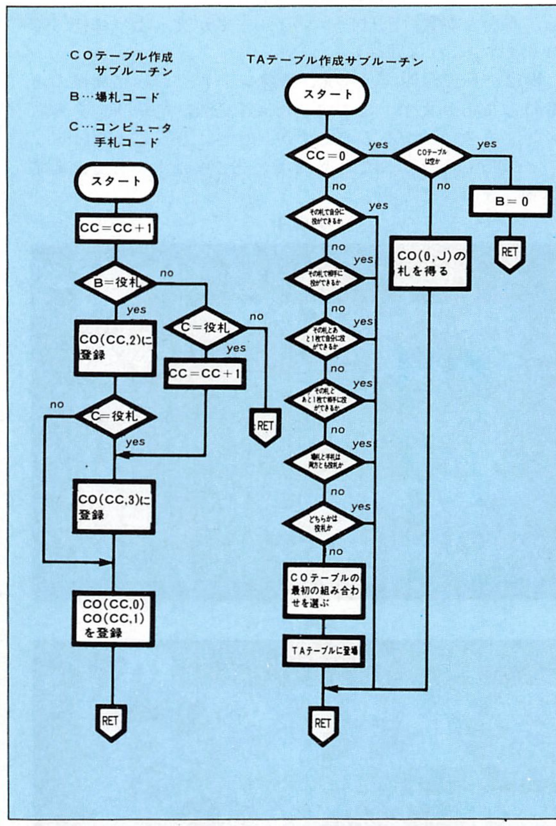


図２



BASICプログラム・リスト

```
0 CLEAR100,&H34E0:DEFINTA-Z:DEFUSR0=&H37E6:DEFUSR1=&H3966:DEFUSR2=&H3572:WIDTH80
,25:CONSOLE24,1,0,0:COLOR0,4
100 DIM A(564),TA(6,2),CO(10,3),F(4,26),F1(48),F2(23),F3(11)
110 DEFFNZ(X)=X-INT(X/12)*12
120 FOR A=0TO564:A(A)=0:NEXT:SS=0
210 DATA0,0,0,0,0,0,5,4,3,3,3,3,5,4,3,3,3,3
820 RANDOMIZE(VAL(RIGHT$(TIME$,2))):OYA=INT(RND(1)*2)
915 CLS:ST=ST+1:SYMBOL(464,8),STR$(ST),2,2
920 COL=6:XX=208:YY=0:FD=24:PT=192:AD=&H39E0:GOSUB10000:XX=336:PT=96:AD=&H3AA0:G
OSUB10000:XX=400:AD=&H3AA0:GOSUB10000
930 COL=2:XX=8:YY=8:FD=16:PT=96:AD=&H3B00:GOSUB10000:YY=72:AD=&H3B60:GOSUB10000
940 FORI=0TO7:SYMBOL(I*56+8,184),STR$(I+1),2,1:NEXT
1020 FORI=1TO48:F1(I)=I:NEXT
1035 RESTORE:FORY=0TO2
1036 FORX=1TO6:READTA(X,Y):NEXTX,Y
1050 IFOYA=0THEN1100
1060 K=0:GOSUB1210:K=1:GOSUB1210:K=2:GOSUB1210:GOTO1140
1100 K=1:GOSUB1210:K=0:GOSUB1210:K=2:GOSUB1210
1140 A=0:FORI=1TO48:IFF1(I)=0 THEN1160
1150 F2(A)=F1(I):A=A+1
1160 NEXT
1170 FORI=0TO50:A=INT(RND(1)*24):B=INT(RND(1)*24)
1180 SWAPF2(A),F2(B):NEXT
1190 GOTO1300
1210 FORJ=0TO11:F3(J)=0:NEXT
1220 FORI=1TO8
```

```
1230 A=INT(RND(1)*48)+1:IFF1(A)=0THEN1230
1240 B=FNZ(A):F3(B)=F3(B)+1:IFF3(B)>2THENF3(B)=F3(B)-1:GOTO1230
1250 F(K,I)=A:F1(A)=0
1260 NEXT:RETURN
1300 FORI=0TO2:X=I:GOSUB6010:NEXT
1330 KAI=0:HM=8:Z=8:GOSUB7080:PB=8:GOSUB7170
1340 COL=2:XX=448:YY=144:FD=47:PT=188:AD=&H3BC0:GOSUB10000
1350 SYMBOL(112,8),STR$(P0)+"ﾓﾝ",2,2
1360 SYMBOL(112,72),STR$(P1)+"ﾓﾝ",2,2
1500 PC=0:PM=0:HC=8:SC=0:SM=0
1505 FORK=3TO4:FORI=0TO24:F(K,I)=0:NEXTI,K
1508 TA(0,0)=9:TA(0,1)=9:TA(0,2)=9
1510 IFOYA=1THEN1800
1520 KAI=KAI+1:IFHC=0THEN8110
1530 GOSUB5010:XC=XA:XD=XB
1540 IFB=0THENPB=PB+1:F(2,PB)=F(0,1):F(0,1)=99:GOTO1570
1550 F(3,PC+1)=C:F(3,PC+2)=B:PC=PC+2:X=3:FL=PC:GOSUB6020
1560 F(2,CO(J,1))=99:X=2:FL=PB:GOSUB6020:PB=PB-1
1565 F(0,CO(J,0))=99
1570 X=0:FL=HC:GOSUB6020:HC=HC-1:F=3:Z=PC:FS=PC:GOSUB7010:Z=PB:GOSUB7170
1580 F=0:GOSUB6110
1590 IFB=0 THENPB=PB+1:F(2,PB)=C:X=2:FL=PB:GOSUB6020:GOTO1620
1600 F(3,PC+1)=C:F(3,PC+2)=B:PC=PC+2:X=3:FL=PC:GOSUB6020
1610 F(2,CO(J,1))=99:X=2:FL=PB:GOSUB6020:PB=PB-1
1620 Z=PB:GOSUB7170:IFFS<>PC THENF=3:Z=PC:GOSUB7010
1630 IFTA(XA,1)=0ANDTA(XA,0)<4 THENTA(XA,0)=4:GOTO1690
1635 IFTA(XB,1)=0ANDTA(XB,0)<4 THENTA(XB,0)=4:GOTO1690
1640 IFTA(XC,1)=0ANDTA(XC,0)<4 THENTA(XC,0)=4:GOTO1690
1645 IFTA(XD,1)=0ANDTA(XD,0)<4 THENTA(XD,0)=4:GOTO1690
1650 GOTO1800
1690 IFSM=1THENA=4:GOTO7510
1700 FORXT=0TO5
1710 IFTA(XT,2)=1ANDTA(XT,0)=2THENA=4:GOTO7510
1720 NEXT
1730 SC=1:A=4:GOSUB8010
1800 KAI=KAI+1:IFHM=0THEN8110
1805 SYMBOL(0,192),"ﾄﾞ ﾚ ｦ ｷﾘﾏｽｶ",2,1:LOCATE30,24:INPUTS:IFS=0THEN1805
1810 IFS>HM THEN1805
1820 C=F(1,S):GOSUB6510:XC=XA:XD=XB
1900 F(1,S)=99:X=1:FL=HM:GOSUB6020:HM=HM-1:GOSUB7080
1910 LOCATE0,23:PRINTSPC(55):FORI=1TOHM:SYMBOL(I*64-56,184),STR$(I),2,1:NEXT
1920 F=4:FS=PM:Z=PM:GOSUB7010:Z=PB:GOSUB7170
1930 C=F2(KAI-1):XX=448:YY=144:AD=&H3C80+360*(C-1):GOSUB20000
1940 GOSUB6510:Z=PB:GOSUB7170:IFFS<>PM THEN F=4:Z=PM:GOSUB7010:Z=PB:GOSUB7170
2000 IFTA(XA,2)=0ANDTA(XA,0)<4THENTA(XA,0)=5:GOTO2020
2005 IFTA(XB,2)=0ANDTA(XB,0)<4THENTA(XB,0)=5:GOTO2020
2010 IFTA(XC,2)=0ANDTA(XC,0)<4THENTA(XC,0)=5:GOTO2020
2012 IFTA(XD,2)=0ANDTA(XD,0)<4THENTA(XD,0)=5:GOTO2020
2015 GOTO1520
2020 SYMBOL(0,192),"ｼｮｳﾌﾞ= 0 ,ｺｲｺｲ = 1",2,1:LOCATE40,24:INPUTSM
2040 IFSM=0THENA=5:GOTO7510
2050 IFSM=1THENSC=0:A=5:GOSUB8010:GQTO1520
2060 GOTO2020
5010 CC=0:CO(0,0)=0
5020 FORJ0=1TOHC
5030 C=F(0,J0)
5040 FORJ2=1TOPB
5050 B=F(2,J2):IFFNZ(B)=FNZ(C)THENGOSUB5510
5060 NEXTJ2,J0
5080 GOSUB5710
5090 RETURN
5510 IFB<28OR(B>33ANDB<39)ORB=42ORB=43THEN5600
5520 CC=CC+1:CO(CC,2)=0:CO(CC,3)=0
5530 IFB=48THENA=1:GOTO5580
5540 IFB>43THENA=3:GOTO5580
5550 IFB>38ANDB<42THENA=6:GOTO5580
5560 IFB>30ANDB<34THENA=5:GOTO5580
5570 IFB>27ANDB<31THENA=4
5580 CO(CC,2)=A:GOTO5620
5600 IFC<28OR(C>33ANDC<39)ORC=42ORC=43THENCO(0,0)=J0:CO(0,1)=J2:RETURN
5610 CC=CC+1:CO(CC,2)=0:CO(CC,3)=0
5620 IFC=48THENA=1:GOTO5670
5630 IFC>43THENA=3:GOTO5670
5640 IFC>38ANDC<42THENA=6:GOTO5670
5650 IFC>30ANDC<34THENA=5:GOTO5670
```

```
5660 IFC>27ANDC<31THENA=4:GOTO5670
5665 GOTO5680
5670 CO(CC,3)=A
5680 CO(CC,0)=J0:CO(CC,1)=J2:RETURN
5710 IFCC=0THEN5900
5715 FORI=1TO2:FORJ=1TOCC
5725 XA=CO(J,2):XB=CO(J,3):T1=TA(XA,0):T2=TA(XB,0):T3=TA(XA,1):T4=TA(XB,1):T5=TA
(XA,2):T6=TA(XB,2)
5730 IFT3>I ANDT4>I THEN 5745
5735 IF(XA=3ANDT5=2)OR(XB=3ANDT6=2)THEN5825
5740 IFT1=1ANDT2=1THEN5825
5745 NEXT
5750 FORJ=1TOCC
5755 XA=CO(J,2):XB=CO(J,3):T1=TA(XA,0):T2=TA(XB,0):T3=TA(XA,1):T4=TA(XB,1):T5=TA
(XA,2):T6=TA(XB,2)
5760 IFT5>I ANDT6>I THEN5775
5765 IF(XA=3ANDT3=2)OR(XB=3ANDT4=2)THEN5825
5770 IFT1=2ANDT2=2THEN 5825
5775 NEXT:NEXT
5780 FORJ=1TOCC
5785 XA=CO(J,2):XB=CO(J,3):T1=TA(XA,0):T2=TA(XB,0)
5790 IFT1=0ANDT2=0THEN5825
5795 NEXT
5800 FORJ=1TOCC
5805 XA=CO(J,2):XB=CO(J,3):T1=TA(XA,0):T2=TA(XB,0)
5810 IFT1=0ORT2=0THEN5825
5815 NEXT
5820 J=1:XA=CO(J,2):XB=CO(J,3):T1=TA(XA,0):T2=TA(XB,0)
5825 IFXA=0THEN5855
5830 IFTA(XA,1)<>0THEN5840
5835 XA=2:TA(XA,1)=1:IF TA(1,1)=4 THEN XA=1:TA(XA,1)=1
5840 TA(XA,1)=TA(XA,1)-1
5845 IFT1=0THENTA(XA,0)=1
5850 IFT1=2THENTA(XA,0)=3
5855 IFXB=0THEN5885
5860 IFTA(XB,1)<>0THEN5870
5865 XB=2:TA(XB,1)=1:IFTA(1,1)=4THENXB=1:TA(XB,1)=1
5870 TA(XB,1)=TA(XB,1)-1
5875 IFT2=0THENTA(XB,0)=1
5880 IFT2=2THENTA(XB,0)=3
5885 B=F(2,CO(J,1)):C=F(0,CO(J,0))
5890 RETURN
5900 IFCO(0,0)=0THENB=0:RETURN
5910 J=0:GOTO5885
6010 FL=8
6020 IFFL<=1THENRETURN
6030 MX=FL-1:FL=0
6040 FORJJ=1TOMX
6050 IFF(X,JJ)>F(X,JJ+1)THENSWAPF(X,JJ),F(X,JJ+1):FL=JJ
6060 NEXT
6070 GOTO6020
6110 C=F2(KAI-1):XX=448:YY=144:AD=&H3C80+(C-1)*360:GOSUB20000
6120 CC=0:CO(0,0)=0:J0=1
6125 IFPB=0THEN6160
6130 FORJ2=1TOPB
6140 B=F(2,J2):IFFNZ(B)=FNZ(C)THENGOSUB5510
6150 NEXT
6160 GOSUB5710:C=F2(KAI-1)
6180 XX=448:YY=144:FD=47:PT=188:AD=&H3BC0:GOSUB10000
6190 RETURN
6510 CC=0:CO(0,0)=0
6515 IFPB=0THEN6550
6520 FORJ2=1TOPB
6530 B=F(2,J2):IFFNZ(B)=FNZ(C)THENGOSUB5510
6540 NEXT
6550 IFCC<>0THEN6580
6560 IFCO(0,0)=0THENXA=0:XB=0:PB=PB+1:F(2,PB)=C:X=2:FL=PB:GOSUB6020:GOTO6770
6570 J=0:B=F(2,CO(J,1)):XA=CO(0,2):XB=CO(0,3):GOTO6740
6580 IFCC=1ORCO(1,1)=CO(2,1)THENJ=1:B=F(2,CO(1,1)):GOTO6690
6590 SYMBOL(0,192),"ﾅﾝﾃﾝﾌﾀﾞ ﾄ ｱﾜｾﾏｽｶ",2,1:LOCATE32,24:INPUTA
6610 IFA=1THENMN=1:MX=23:MM=35:GOTO6660
6620 IFA=5THENMN=24:MX=33:MM=0:GOTO6660
6630 IFA=10THENMN=34:MX=43:MM=0:GOTO6660
6640 IFA=20THENMN=44:MX=48:MM=0:GOTO6660
6650 GOTO6590
```

```
6660 FORJ=1TOCC
6670 B=F(2,CO(J,1)):IF(B>=MN ANDB=<MX)ORB=MM THEN6690
6680 NEXT:GOTO6590
6690 XA=CO(J,2):XB=CO(J,3)
6695 IFXA=0THEN6720
6700 IFTA(XA,2)<>0THEN6710
6705 XA=2:TA(XA,2)=1:IFTA(1,2)=4THENXA=1:TA(XA,2)=1
6710 TA(XA,2)=TA(XA,2)-1
6715 IFTA(XA,0)<2THENTA(XA,0)=TA(XA,0)+2
6720 IFXB=0THEN6745
6725 IFTA(XB,2)<>0THEN6735
6730 XB=2:TA(XB,2)=1:IFTA(1,2)=4THENXB=1:TA(XB,2)=1
6735 TA(XB,2)=TA(XB,2)-1
6740 IFTA(XB,0)<2THENTA(XB,0)=TA(XB,0)+2
6745 PM=PM+1:F(4,PM)=B:PM=PM+1:F(4,PM)=C:X=4:FL=PM:GOSUB6020
6750 F(2,CO(J,1))=99:X=2:FL=PB:GOSUB6020:PB=PB-1
6770 XX=448:YY=144:FD=47:PT=188:AD=&H3BC0:GOSUB10000
6780 RETURN
7010 DY=64*(F-3)+24:LINE(0,DY)-(511,DY+47),PRESET,,BF
7020 IFZ=0THENRETURN
7030 IFZ<9THENX1=32:GOTO7050
7040 X1=215/(Z-1)
7050 FORX=1TOZ:I=F(F,X):XX=2*X1*(X-1):YY=DY:AD=&H3C80+360*(I-1):GOSUB20000:NEXT:
RETURN
7080 LINE(0,136)-(447,183),PRESET,,BF
7100 IFHM=0THENRETURN
7110 IFHM=8THENX1=27:GOTO7130
7120 X1=32
7130 FORX=1TOHM:I=F(1,X):IFI=0THEN7150
7140 XX=2*X1*(X-1):YY=136:AD=&H3C80+(I-1)*360:GOSUB20000
7150 NEXT:RETURN
7170 LINE(512,0)-(639,199),PRESET,,BF
7180 IFZ=0THENRETURN
7210 Z1=INT(Z/2):Z2=Z-Z1
7220 IFZ1=0THEN7275
7230 IFZ1=1THENY1=48:GOTO7250
7240 Y1=144/(Z1-1):IFY1>48THENY1=48
7250 FORY=1TOZ1:I=F(2,Y):XX=512:YY=Y1*(Y-1):AD=&H3C80+(I-1)*360:GOSUB20000:NEXT
7275 IFZ2=1THENY2=48:GOTO7290
7280 Y2=144/(Z2-1):IFY2>48THENY2=48
7290 FORY=1TOZ2:I=F(2,Y+Z1):XX=576:YY=Y2*(Y-1):AD=&H3C80+(I-1)*360:GOSUB20000:NE
XT:RETURN
7510 AD=&H3538:SS=USR2(AD)
7515 FORI=0TO3:B=(A-4)*64+8:LINE(8,B)-(103,B+15),PRESET,,BF:XX=8:YY=B:FD=16:PT=9
6:AD=&H3B00+(A-4)*96:GOSUB10000
7540 COL=3:GOSUB10000
7560 NEXT
7570 LINE(512,0)-(639,199),PRESET,,BF
7580 LOCATE65,1:PRINT" ** ﾄｸﾃﾝ **":DY=3:P=0
7610 IFTA(1,0)=A THENP=P+15:LOCATE65,DY:PRINT"ｺﾞ  ｺｳ..15ﾓﾝ":DY=DY+2:GOTO7625
7615 IFTA(2,0)=A THENP=P+10:LOCATE65,DY:PRINT"ｼ   ｺｳ..10ﾓﾝ":DY=DY+2:GOTO7625
7620 IFTA(3,0)=A THENP=P+8:LOCATE65,DY:PRINT"ｻﾝ ｺｳ..8ﾓﾝ":DY=DY+2
7625 IFTA(4,0)=A THENP=P+6:LOCATE65,DY:PRINT"ｱｵ ﾀﾝ..6ﾓﾝ":DY=DY+2
7630 IFTA(5,0)=A THENP=P+6:LOCATE65,DY:PRINT"ｱｶ ﾀﾝ..6ﾓﾝ":DY=DY+2
7635 IFTA(6,0)=A THENP=P+5:LOCATE65,DY:PRINT"ｲﾉｼｶﾁｮｳ.5ﾓﾝ":DY=DY+2
7640 IFA=4THENP0=P0+P ELSEP1=P1+P
7660 LINE(112,8)-(205,24),PRESET,,BF:SYMBOL(112,8),STR$(P0)+"ﾓﾝ",2,2
7670 LINE(112,72)-(205,88),PRESET,,BF:SYMBOL(112,72),STR$(P1)+"ﾓﾝ",2,2
7672 IFST=12THENLOCATE0,20:PRINT"ｼｭｳﾘｮｳﾃﾞｽ｡";:ENDELSELOCATE0,24:PRINT"Push Any k
EY";
7676 IFINKEY$=""THEN7676
7680 IFA=4THENOYA=0:GOTO915
7690 IFA=5THENOYA=1:GOTO915
8010 AD=&H34F3:SS=USR2(AD)
8020 FORI=0TO3:B=(A-4)*64+8:LINE(8,B)-(103,B+15),PRESET,,BF:XX=8:YY=B:FD=16:PT=9
6:AD=&H3B00+(A-4)*96:GOSUB10000:NEXT:RETURN
8110 IFSC=1THENA=4:GOTO7570
8120 IFSM=1THENA=5:GOTO7570
8130 IFOYA=0THENP=6:P0=P0+P:A=5:GOTO7660
8140 P=6:P1=P1+P:A=4:GOTO7660
10000 POKE &H39D8,(VARPTR(A(0))¥ 256):POKE &H39D9,(VARPTR(A(0)) MOD 256):POKE &H
39D7,PT:SS=USR1(AD):LINE(XX,YY)-(XX+16*PT/FD-1,YY+FD-1),PSET,0,BF:PUT@(XX,YY)-(X
X+16*PT/FD-1,YY+FD-1),A,PSET,COL:RETURN
20000 POKE &H39D8,(VARPTR(A(0))¥ 256):POKE &H39D9,(VARPTR(A(0)) MOD 256):SS=USR0
(AD):PUT@A(XX,YY)-(XX+63,YY+46),A,PSET:RETURN
```

ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
34F0 30 34 00 33 33 33 40 28 28 28 28 40 22 22 28 28 :B1
---------------------------------------------------------
Sum: 30 34 00 33 33 33 40 28 28 28 28 40 22 22 28 28 :B1
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3500 28 33 33 33 40 22 22 28 28 2D 2D 33 33 2D 2D 28 :D7
3510 28 22 22 22 22 40 28 28 28 28 40 22 22 22 22 40 :9B
3520 19 19 13 13 16 16 19 19 40 22 22 40 22 1E 1E 1E :F6
3530 22 22 19 40 19 40 19 FF 1E 22 28 22 1E 1E 1E 1E :10
3540 1B 40 1E 22 28 22 2D 2D 2D 2D 40 28 28 28 22 28 :9B
3550 2D 33 2D 28 2D 33 3D 3D 40 40 1E 1E 1E 40 40 1E :07
3560 1E 40 22 40 22 40 1E 40 22 40 1E FF 1E FF 1E 1E :58
3570 00 FF AE 02 E6 84 8D 36 E6 84 86 03 3D E7 8D 00 :80
3580 45 E6 8D 00 41 A6 84 81 40 27 0A 81 FF 26 01 39 :F5
3590 86 8F B7 FD 03 5A 26 FD E6 8D 00 2A 86 0F B7 FD :2F
35A0 03 5A 26 FD 6A 8D 00 1D 26 D7 30 01 20 C6 86 FF :2D
35B0 E7 8D 00 11 5F 5C A0 8D 00 0B 24 F9 5A 86 11 3D :C3
35C0 E7 8D 00 01 39 11 FD 7F FF FF FF 7F FF FF FB 6F :1F
35D0 DF 7D BF 7D C0 03 FF 7F C0 03 EF 7F 80 03 FD 7F :09
35E0 80 07 FF 77 B8 1F FF 7F FE 3F FF 6F FF FF F3 7F :6D
35F0 FE 00 73 7F FE FF 61 7B FE FF 41 70 FE FF 41 70 :25
---------------------------------------------------------
Sum: EA AF 37 B3 AA EC 37 68 2A A0 45 81 B1 5A 13 57 :BD
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3600 7E FE 43 70 05 FE 97 60 05 FE BF 60 05 FE FF 70 :BD
3610 05 FE 8F 78 0D FC 87 68 0B FD 87 78 0B FD 03 7C :90
3620 1B FD 01 56 3B FD 01 7F F3 F9 01 7F F7 F8 87 71 :7A
3630 F7 F8 7F 70 27 F0 0D 60 2F F0 01 40 2F F0 01 40 :22
3640 4F E0 01 40 5F E0 01 40 BF C0 01 60 BF C0 01 7F :CF
3650 3F 80 01 7E 7F 00 01 7C 00 01 81 7F FC 00 61 7F :17
3660 F8 00 3F 77 F8 00 07 6D FC 00 07 7F FC 00 07 7B :1A
3670 DE 00 3F 6F 7F 80 FF 7F FF E7 FF 7F 70 0F 80 01 :6D
3680 70 0F 80 01 70 1F 80 C1 78 1F C0 79 7F 7F C0 FF :5D
3690 7E FF FF 87 5F FF FE 3F 4F FF FE 7F 47 FF FC 7F :2A
36A0 43 FF F8 FF 67 FF C4 FF 6C 7F BE FF 79 3F 7D FF :3E
36B0 7B 9A FD FF 77 C4 FD FF 77 FF FD FF 77 FF FD FF :2C
36C0 50 FF A3 FF 4E 1F 3F FF 67 CE 43 FF 66 70 83 FF :6B
36D0 73 1D 87 FF 71 03 86 1F 71 83 86 1F 70 81 06 0F :CE
36E0 40 81 06 0F 40 01 E4 0F 40 02 34 07 40 02 1C 27 :0C
36F0 40 02 10 23 40 02 00 33 40 8C 00 33 70 04 00 0B :68
---------------------------------------------------------
Sum: E8 97 86 08 B5 4D 1C AD EE 07 46 C2 99 65 4E D3 :F4
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3700 70 00 00 01 70 00 00 0F 70 00 00 3D 70 00 00 FD :0A
3710 70 00 01 F9 60 00 01 F9 40 00 01 F1 40 00 81 E1 :98
3720 40 03 81 E1 40 0F 81 C1 40 3F 81 C1 40 FF 81 81 :38
3730 76 40 7F F0 01 40 FF F8 01 47 E3 FE 01 7F E3 FF :E8
3740 C1 7F 03 FF F9 7F 01 FF FF 78 00 7F FF 78 00 1F :46
3750 FF 78 00 1F FF 60 00 07 FF 68 00 03 FF 68 00 0B :D8
3760 FF 68 00 09 FF 68 00 FC FF 68 00 FE 7F 48 00 FF :FE
3770 3F 7C 1F FE 3F 5F FF FC 07 4F FF F8 71 4F FF F0 :6D
3780 E1 47 FF E3 C1 47 FB FF 83 43 F1 FE 07 61 FB F8 :1C
3790 0F 78 F9 E0 3F 7F FB C3 FF 7F FF FF 3F 7F FF FE :13
37A0 07 7F FF FC 01 7F FF FC 01 7F FB FC 01 7F FC F0 :DF
37B0 01 60 30 00 01 40 00 00 01 40 00 00 01 40 00 00 :54
37C0 01 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 :04
37D0 01 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 :04
37E0 01 40 00 00 01 7E 34 77 10 AE 02 AE 8D 01 E9 86 :D6
37F0 03 CE 3F FF EF 84 EF 89 01 70 EF 89 01 78 EF 89 :D4
---------------------------------------------------------
Sum: 92 4A 89 AE 3B FC 99 7D 8C 3C 40 95 B7 8D B2 6C :5F
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3800 02 E8 EF 89 02 F0 EF 89 04 60 30 02 CE FF FF EF :1D
3810 84 EF 89 01 70 EF 89 01 78 EF 89 02 E8 EF 89 02 :3A
3820 F0 EF 89 04 60 30 02 4A 26 E5 A6 A4 10 2B 00 AC :84
3830 A6 24 10 2B 00 CE A6 28 10 2B 01 0E A6 2C 10 2B :F8
3840 01 69 A6 A8 10 10 2B 01 6B A6 A8 14 10 2B 01 6D :7A
3850 A6 A8 18 2B 29 8D 4B 34 10 30 89 01 88 8D 43 35 :1D
3860 10 30 88 10 86 02 C6 B4 34 06 A6 89 FE 88 AA 89 :FC
3870 01 78 A7 80 35 06 5A 26 EF 4A 26 EA 35 F7 8D 22 :7F
3880 30 88 10 8D 1D 30 88 10 86 02 C6 B4 34 06 A6 89 :A5
3890 FD 10 A4 89 FE 88 A7 80 35 06 5A 26 EF 4A 26 EA :EB
38A0 35 F7 C6 B4 A6 A0 34 24 10 8E 00 00 C6 08 48 34 :2C
38B0 07 24 02 31 21 1F 20 10 AF 84 E3 84 1F 02 35 07 :C5
38C0 34 07 24 02 31 21 1F 20 10 AF 84 E3 84 1F 02 35 :F2
38D0 07 5A 26 DA 35 24 30 02 5A 26 C9 39 34 10 30 89 :6B
38E0 01 78 8D BE 35 10 34 10 8D B8 35 10 31 89 02 F0 :83
38F0 86 02 C6 B4 34 06 A6 80 A7 A0 35 06 5A 26 F5 4A :A3
---------------------------------------------------------
Sum: FF 31 17 65 77 54 62 81 68 CC 17 CE 82 B4 85 BB :E9
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3900 26 F0 35 F7 34 10 30 89 01 78 34 10 8D 94 30 88 :D5
3910 10 34 10 8D 8D 35 10 35 20 35 40 34 30 86 02 C6 :2F
3920 B4 34 06 A6 A0 43 A4 80 A7 C0 35 06 5A 26 F2 4A :F9
3930 26 ED 35 30 86 02 C6 B4 34 06 A6 A4 AA 80 A7 A0 :6F
3940 35 06 5A 26 F3 4A 26 EE 35 F7 34 10 30 89 01 78 :AE
3950 17 FF 4F 35 10 34 10 17 FF 48 35 10 31 89 01 78 :C4
3960 33 89 02 F0 20 B5 34 37 10 AE 02 AE 8D 00 69 A6 :F8
3970 A0 34 20 10 8E 00 00 C6 08 48 34 07 24 02 31 21 :5B
3980 1F 20 10 AF 84 E3 84 1F 02 35 07 34 07 24 02 31 :D8
3990 21 1F 20 10 AF 84 E3 84 1F 02 35 07 5A 26 DA 35 :F6
39A0 20 30 02 6A 8D 00 30 26 C6 35 B7 34 20 10 AE 8D :F0
39B0 00 28 20 10 34 20 10 AE 8D 00 21 20 07 34 20 10 :A3
39C0 AE 8D 00 1A 17 FE DB 35 20 30 88 10 17 FE D3 30 :7A
39D0 88 10 17 FE CD 35 F7 00 2C 24 35 C7 36 7C 37 31 :0C
39E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04 00 :04
39F0 02 01 00 08 00 00 0A 00 06 03 00 0C 0E 10 84 00 :CC
---------------------------------------------------------
Sum: C7 3C B4 0E 70 77 97 A0 0E 6B BF 35 B6 EC A3 53 :E8
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3A00 76 03 B0 0C 07 13 E0 00 3F DF E0 0C 07 10 80 00 :D0
3A10 0F 8F C0 1C 06 10 80 00 06 03 80 1C 04 10 80 00 :49
3A20 02 01 80 3F 86 08 80 00 00 B1 07 FF C6 08 88 00 :DD
3A30 01 B8 33 FF 04 00 08 00 07 38 70 1C 0C 00 08 00 :D6
3A40 1F 18 60 0C 08 00 08 00 7F 0C C0 00 04 30 08 00 :3A
3A50 7B 0E 00 04 04 38 08 40 72 0C 00 0C 0C 18 03 F0 :B2
3A60 06 18 00 1E 0C 1C 00 44 06 18 08 7F 0C 0C 00 42 :A7
3A70 06 18 09 EF 8C 0C 01 C0 0C 18 0D CC 0C 0C 01 40 :C5
3A80 0C 18 1D 8C 0C 1C 01 C2 0C 19 FC 0C 0F FC 00 42 :32
3A90 0C 0F FC 0C 07 F8 00 3E 00 00 00 00 00 00 00 00 :60
3AA0 00 00 00 00 60 00 00 00 3F 80 00 00 1F C0 00 00 :FE
3AB0 03 80 00 00 03 00 00 00 06 00 00 00 00 00 00 00 :8C
3AC0 00 04 00 00 00 07 00 00 00 07 80 00 00 07 80 60 :79
3AD0 10 03 80 78 20 03 80 3C 60 03 80 3E F0 33 80 1E :CC
3AE0 FF F1 80 1E 7F E1 C2 1E 0F C0 C6 0C 00 00 EC 00 :5B
3AF0 00 00 7C 00 00 00 3C 00 00 00 18 00 00 00 00 00 :D0
---------------------------------------------------------
Sum: 58 40 21 B1 50 8A 78 9E 0F 76 86 F0 23 7E 88 32 :B0
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3B00 FF FF FF FF FF FF FF FF FF 87 FF FF FF FF FD BB :32
3B10 87 9F 83 FB EC 43 E7 9F C1 F3 0C 3F EC 0F F1 E2 :26
3B20 1C 8F FC 0F E7 E6 DC DF 87 9F FF EF 9C FD C3 B1 :5F
3B30 FF CF 9E 7D F3 39 DF CF BE 58 F3 3B 9F C3 9E 55 :5C
3B40 73 2F 81 93 CC D5 73 21 C0 99 61 CD 67 21 E1 98 :73
3B50 F3 CD EF 7F FF FF FF FF FF FF FF FF FF FF FF FF :22
3B60 FF FF FF FF FF FF FF 7F F7 FF EF FF FB 33 D3 FF :5C
3B70 67 FF F8 03 C0 47 00 FF FC 03 C0 E3 01 FF FF 3F :47
3B80 F3 F3 E7 FF FE 03 F3 F7 E6 1F FC 19 F3 EF E7 9F :39
3B90 F8 B4 E7 EF C7 BF F3 8C E7 07 CF FF F1 9C E6 63 :19
3BA0 CE FF F8 3C E6 E5 CC F7 FC 79 FE 47 DC 07 FF E3 :0E
3BB0 FF 0F FE 0F FF FF FF FF FF FF FF FF FF FF FF FF :0F
3BC0 7F FF FF FF 7F FF FF FF 70 00 00 03 77 FF FF FB :DB
3BD0 77 FF FF FB 77 FF FF FB 77 FF FF FB 77 FF FF FB :C0
3BE0 77 FF FF FB 77 FF FF FB 77 FF FF FB 77 FF FF FB :C0
```

```
3BF0 77 FF FF FB 77 FF FF FB 77 FF FF FB 77 FF FF FB :C0
-------------------------------------------------------
Sum: 09 A7 43 C3 DD 1D BF 53 54 A6 D1 68 23 AD C8 48 :D5


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3C00 77 FF FF FB 77 FF FF FB 77 FF FF FB 77 FF FF FB :C0
3C10 77 FF FF FB 77 FF FF FB 77 FF FF FB 77 FF FF FB :C0
3C20 77 FF FF FB 77 FF FF FB 77 FF FF FB 77 FF FF FB :C0
3C30 77 FF FF FB 77 FF FF FB 77 FF FF FB 77 FF FF FB :C0
3C40 77 FF FF FB 77 FF FF FB 77 FF FF FB 77 FF FF FB :C0
3C50 77 FF FF FB 77 FF FF FB 77 FF FF FB 77 FF FF FB :C0
3C60 77 FF FF FB 77 FF FF FB 77 FF FF FB 77 FF FF FB :C0
3C70 70 00 00 03 7F FF FF FF 7F FF FF FF 00 00 00 00 :6B
3C80 7F FE FF FF 7D FD 66 E7 7C 7B 83 43 7C 77 A7 27 :C0
3C90 76 2C BC 7F 71 88 50 33 71 84 C1 A1 7A 43 93 93 :93
3CA0 7E E3 1F DF 7C FB 1C 9F 7E FF B8 5F 7E FF FC FF :9D
3CB0 7E FF BF FF 7E FF BF FF 7E FF BF F7 7E FF BF FF :E4
3CC0 7E FF BF FF 7E FF 3F FF 7E FF 9F FF 7C FF BF EF :3A
3CD0 7E FF BF EF 7E FF BF EF 7E FF 3F F7 7E FF 9F F7 :1C
3CE0 7E FF BF F7 7C FF BF FB 7E FF BF CB 7E FF 3F BB :E6
3CF0 7E FF 9F 0D 7F FF BF 0D 7F FF BF 7D 7F FF BF 9B :05
-------------------------------------------------------
Sum: 9A 01 6E 2E 9F 73 05 8A A2 F1 0F 54 2A AD 4A D1 :C0


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3D00 7F FF 3F E7 7F FF 9F DF 7F FF FF DF 7F FF FF BF :38
3D10 7F FF FF A7 7F FF FF BB 7F FF FF DB 7F FF FF E7 :18
3D20 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
3D30 7F FF FF FF 7F FE FF FF 7D FD 66 E7 7C 7B 83 43 :7B
3D40 7C 77 A7 27 76 2C BC 7F 71 88 50 33 71 84 C1 A1 :71
3D50 7A 43 13 93 7E E3 1F DF 7C 7B 1C 9F 5A 3F B8 5F :24
3D60 66 3F FC E3 46 BF BF E1 4E FF BF 2B 7E FF BE 17 :B2
3D70 7E F9 BF 37 7E 90 B3 E3 66 09 21 E3 42 9F 93 F7 :EF
3D80 64 FF BF EF 7E F9 BF EF 7E 90 B3 EF 7E 09 21 F7 :85
3D90 66 9F 93 F7 42 FF BF F7 64 F9 BF FB 7E 90 B3 CB :29
3DA0 7E 09 21 BB 7E 9F 93 0D 7C 7F BF 0D 7C 79 BF 7D :18
3DB0 7E F0 B3 9B 7F F9 21 E7 7F FF 93 DF 7F FF BF DF :48
3DC0 7F FF 1F BF 7F FF 1F A7 7F FF BF BB 7F FF FF DB :F0
3DD0 7F FF FF E7 7F FF FF FF 7F FF FF FF 7F FF FF FF :D8
3DE0 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
3DF0 7F FF FF E1 7F FF FF 83 7F FF FF 8F 7F FF FF 1F :06
-------------------------------------------------------
Sum: 98 81 F3 22 6D E5 37 BC 74 08 2F 9E 77 E6 98 0C :BD


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3E00 7F FF FE 1F 4F FF FE 3F 63 FF FC 3F 71 FF FC 7F :AE
3E10 78 FF F8 7F 7C FF F8 7F 7C 7F FC 7F 7C 7F F2 7F :C2
3E20 7C FF ED FF 7D 9F DE FF 7F 6F DE FF 7F 70 2F 7F :C8
3E30 7E EF F7 7F 7E C7 E8 7D 7E 3B EF 31 7C ED D9 E3 :8B
3E40 4D 8E B1 C7 67 8A 43 8F 63 8A C3 1F 71 8A C2 1F :C1
3E50 71 C2 C0 3F 78 C2 C0 7F 78 42 80 71 5C 47 80 E3 :5C
3E60 4C 03 80 C7 46 03 80 CF 63 03 80 8F 70 83 81 0F :26
3E70 7C 41 00 1D 46 01 00 19 46 00 00 33 63 00 00 43 :59
3E80 71 00 00 07 70 80 00 0F 78 00 00 0F 78 00 00 1F :95
3E90 78 00 00 3F 7C 00 00 3F 7C 00 00 7F 7F FF FF FF :E9
3EA0 7F FF FF FF 7F FF FF E1 7F FF FF 83 7F FF FF 8F :E6
3EB0 7F FF FF 1F 7F FF FE 1F 4F FF FE 3F 63 FF FC 3F :5F
3EC0 71 FF 7C 7F 78 E6 38 7F 7C E7 18 7F 7C 4E DC 7F :9F
3ED0 7C 6C F2 7F 7C FD E1 FF 7D 99 C0 FF 7F 0B C0 FF :D0
3EE0 7F 00 00 7F 7E 00 00 7F 7E 00 08 7D 7E 38 0F 31 :F4
3EF0 7C EC 19 E3 4D 8E 31 C7 67 8A 43 8F 63 8A C3 1F :C9
-------------------------------------------------------
Sum: 46 D5 50 CA DA A3 86 42 00 FF 08 1A 3D 47 21 6E :4E


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3F00 71 8A C2 1F 71 C2 C0 3F 78 C2 C0 7F 78 42 80 71 :32
3F10 5C 47 80 E3 4C 03 80 C7 46 03 80 CF 63 03 80 8F :A9
3F20 70 83 81 0F 7C 41 00 1D 46 01 00 19 46 00 00 33 :36
3F30 63 00 00 43 71 00 00 87 70 80 00 0F 78 00 00 0F :24
3F40 78 00 00 1F 78 00 00 3F 7C 00 00 3F 7C 00 00 7F :04
3F50 FF FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :70
3F60 7F FF FF FF 7F FF FF FF 7F FF FF FF 7B FF FB FF :E8
3F70 7C FF F9 FF 7D FF F3 FF 7E FF FB FF 7E FF FB FF :CF
3F80 7E F7 B9 FF 7F 73 9B FF 7E 71 8B DF 7C 31 C3 9F :21
3F90 7E 1B CB 1B 7F DD EA 1B 7F 80 03 33 78 5F 78 E3 :47
3FA0 70 9C E6 77 79 88 C7 8F 7F 08 87 E3 7F 99 9F F9 :61
3FB0 7F FF FF FD 7F FF BF 0F 7F FF 8E 1F 7F FF 8F 1F :1D
3FC0 7F FD CE BF 7F FC 01 E3 7E E1 FB 87 7F 0F E5 C7 :83
3FD0 7E 37 8E AF 7F BB 0F 71 7F FF 1F B9 7F FF DC D1 :2D
3FE0 7F FF C4 6F 7F FF C6 57 7F FF 27 8B 7F FE 00 9D :96
3FF0 7F FC 3E 1F 7F FC F1 BF 7F FF E3 C7 7F FF FF F9 :A1
-------------------------------------------------------
Sum: F8 2D 81 FA 8F 8C 03 08 62 19 00 58 7B 75 1E 86 :2D


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4000 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
4010 7B FF FB FF 71 FF F1 FF 71 FF F1 FF 7B 9F FB FF :48
4020 7B 0F FA 3F 64 9F 98 1F 41 FF 02 3F 66 FF 9B FF :FD
4030 7E FF FA 7F 7E F7 B8 3F 7F 73 9A 7F 7E 71 8B DF :C6
4040 7C 31 C3 9F 7E 1B CB 1B 7F DD EA 1B 7F 80 03 33 :24
4050 78 5F 78 E3 70 9C E6 77 79 88 C7 8F 7F 08 87 E3 :E3
4060 7F 99 9F F9 7F FF FF FD 7F FD BF 0F 7F C8 8E 1F :68
4070 79 88 8F 1F 78 CD CE BF 78 4C 01 E3 7E E1 FB 87 :0A
4080 7F 0F E5 C7 78 37 8E AF 70 BB 0F 71 79 11 1F B9 :33
4090 7F 11 DC D1 7F BB C4 6F 7F FF C6 57 7F FF 27 8B :75
40A0 7F FE 00 9D 7F FC 3E 1F 7F FC F1 BF 7F FF E3 C7 :45
40B0 7F FF FF F9 7F FF FF FF FF FF FF 7E 7F FF FF :E9
40C0 70 3F FF FF 60 0F 81 3F 70 00 7E 9F 78 07 7D DF :44
40D0 78 0F A3 A1 78 0F 8D 41 7F 08 7C 6F 7F F7 7D AF :34
40E0 7F F7 03 2F 7F F6 00 9F 7F F8 03 FF 7F FE 07 FF :B8
40F0 7F BC 67 FF 7F 9C E3 FF 7C 0C F0 7F 7C 06 F8 0F :1E
-------------------------------------------------------
Sum: C1 DB 23 52 82 B4 3E 04 F6 DF AF 6A C0 CF 54 3E :98


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4100 7E 03 F0 01 6F 03 F0 01 67 80 10 01 63 E3 D0 01 :E4
4110 41 EF C0 01 40 0F DF 01 40 F7 9F 01 41 F8 3E 23 :91
4120 43 F2 B4 E7 43 EA C9 CF 42 08 0D DF 40 62 AD DF :F9
4130 7E 65 48 5F 7E 60 87 01 7E 6C 13 C1 78 41 FB C1 :23
4140 60 01 FB 81 40 02 F8 01 40 3E 00 01 40 7E 00 11 :66
4150 40 7E 00 09 40 FE 00 0F 43 FE 00 3F 7F FE 00 3F :50
4160 7F FF 00 3F 7F FF F0 3F 7F FF FF FF 7F FF FF FF :62
4170 7E 7F FF FF 70 3F FF FF 60 0F 81 3F 70 00 00 1F :66
4180 78 00 00 1F 78 00 00 01 78 00 00 01 7F 00 00 0F :17
4190 7F F0 00 0F 7F F0 00 0F 7F F0 00 1F 7F F8 03 FF :03
41A0 7F FE 07 FF 7F BC 67 FF 7F 9C E3 FF 7C 0C F0 7F :18
41B0 7C 06 F8 0F 7E 03 F0 01 6F 03 F0 01 67 80 10 01 :56
41C0 63 E0 10 01 41 E0 00 01 40 00 00 01 40 00 00 01 :F8
41D0 40 00 00 03 40 00 00 07 40 00 00 0F 40 00 00 1F :38
41E0 40 00 00 1F 7E 00 00 1F 7E 00 00 01 7E 00 00 01 :FA
41F0 78 00 00 01 60 00 00 01 40 00 00 01 40 3E 00 01 :9A
-------------------------------------------------------
Sum: 6A 1A B5 70 32 29 5D 58 EC C4 22 52 29 BB B8 E2 :5B


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4200 40 7E 00 11 40 7E 00 09 40 FE 00 0F 43 FE 00 3F :63
4210 7F FE 00 3F 7F FF 00 3F 7F FF F0 3F 7F FF FF FF :A2
4220 00 00 00 00 80 00 00 00 00 00 00 00 00 00 00 00 :80
4230 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
4240 00 10 00 00 00 10 00 00 00 30 00 00 20 38 00 F0 :98
4250 3E 78 00 7C 3F D8 00 36 1B B8 00 1A 0E F8 00 0C :7E
4260 0F 0C 00 0E 3E F6 00 3C 3B FF 00 E2 3F A0 03 FC :93
4270 39 80 00 3A 30 80 00 16 00 00 00 1C 00 00 00 18 :ED
4280 00 00 00 20 00 00 00 00 00 01 80 00 00 01 C0 00 :62
4290 04 01 E0 00 0E 00 C0 00 1E 00 80 00 00 30 00 00 :81
42A0 00 3C 00 00 08 1C 00 00 0C 08 00 00 07 00 00 00 :7B
42B0 07 80 C0 00 03 CF 80 00 03 7C 00 00 02 AC 00 00 :C6
42C0 27 FE 00 00 3B 67 00 00 39 C0 00 00 01 80 00 00 :41
42D0 01 00 00 00 7E 00 FF FF 7E 00 7F FF 7E 00 3F FF :35
42E0 7C 00 1F FF 7C 01 1F FF 78 43 0F FF 78 E3 CF FF :27
42F0 73 E7 F7 FF 7F C7 FF FF 7F C7 FF FF 4F 87 FE 0F :BB
```

ダンプ・リスト

```
-----------------------------------------------------------
Sum: 67 32 B6 32 B9 F5 5D CD F0 33 7D 63 7E 94 CE 5B :97


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4300 41 83 FE 03 40 03 FE 01 40 03 FF 01 40 03 FF 81 :0D
4310 60 01 FF C1 40 00 FF C1 40 00 7E 01 40 00 78 01 :99
4320 40 00 60 01 40 1F F0 01 42 0F FF C1 47 0F FF C1 :18
4330 7F 00 0F 81 7E 00 1F 83 78 00 7F 0F 60 01 BF 7F :D4
4340 70 01 DF FF 74 01 EF FF 6E 00 CF FF 5E 00 BF FF :0A
4350 60 30 7F FF 66 BC 7F FF 63 DD 7F FF 60 EB FF FF :B5
4360 70 37 0F FF 70 10 1F FF 78 00 1F FF 78 00 7F FF :DF
4370 40 01 FF FF 60 00 7F FF 78 00 3F FF 78 08 7F FF :D1
4380 44 1F FF FF 7C 3F FF FF 00 00 00 00 80 00 00 00 :9A
4390 00 0F E0 00 00 7F FC 00 00 7F FC 00 00 7F FC 00 :60
43A0 00 7F FC 00 00 00 00 00 00 00 00 00 00 00 00 00 :7B
43B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
43C0 00 00 00 00 00 00 00 00 00 00 00 00 00 3F C0 00 :FF
43D0 00 FF F8 00 03 FF FC 00 07 FF FE 00 07 FF FF 00 :FE
43E0 07 FF FF 00 0F FF FF 80 0F FF FF 80 0F FF FF 80 :AC
43F0 07 FF FF 80 07 FF FF 80 03 FF FF 00 03 FF FE 00 :0B
-----------------------------------------------------------
Sum: 32 97 A9 C1 7D AA 0D 41 14 6B 9F 4E 6E C1 A9 3E :2A


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4400 01 FF FE 00 00 FF FC 00 00 7F F0 00 00 3F 80 00 :27
4410 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
4420 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
4430 00 00 00 00 00 00 00 00 00 00 00 00 7F FF FF FF :7C
4440 7F FF FF FF 7F F9 3F FF 7F F9 3B FF 7F F9 33 FF :8E
4450 7F 8A 67 FF 7F C4 4F FF 7F F8 3F FF 7F F8 3F FF :6A
4460 7E 4C 7F FF 7C 04 FF FF 7C 01 81 FF 7C 01 00 FF :3F
4470 78 F0 00 FF 7F 98 01 FF 7E 00 7F FF 7C 01 FF FF :F5
4480 78 3F FF FF 41 FD FF 8F 43 F8 FF 07 47 F8 FF 01 :01
4490 44 FD DF 01 60 7F BF 81 7C FF 7F C1 7F 8F F3 C1 :BD
44A0 58 7F E1 C1 44 F7 F3 C1 47 EF FF C1 47 FF 1F C3 :86
44B0 43 9F 3F EF 43 9F 3F FF 43 9F BF FF 43 BF FF FF :D0
44C0 46 3F FF FF 7C 00 0F FF 78 00 07 FF 78 00 03 FF :05
44D0 7C 00 03 FF 78 80 07 FF 71 80 3F FF 79 80 3F FF :E2
44E0 78 00 7F FF 70 7F FF FF 73 FF FF FF 67 FF FF FF :B7
44F0 FF FF FF FF 7F FF FF FF 6F FF FF FF 67 FF FF FF :48
-----------------------------------------------------------
Sum: 85 5C 61 A8 04 68 8E C8 0C 74 EA 80 84 F4 40 7B :C9


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4500 67 FF FF FF 67 FF FF FF 47 FF FF 7F 67 FF FF 3F :30
4510 47 FF FF 3F 67 FF FF 3F 67 FF FF 27 61 FF F7 1B :26
4520 72 FF F3 1B 72 FF F3 27 71 FF E3 3F 43 FF DB 3F :F8
4530 71 FF DB 3F 71 FF E3 27 41 FF F3 1B 70 FF F3 1B :CF
4540 77 7F F3 27 77 7F F3 3F 4F 8F F3 3F 7D F7 F3 3F :EE
4550 78 F7 F3 3F 72 74 F2 3F 4D 8B 70 7F 6D BB 70 FF :16
4560 72 7C F1 FF 7C FF E3 F1 7B 7F DD C7 77 BF DD C7 :A5
4570 77 BF 3E 19 45 8E F7 DF 7D F6 E3 DF 70 7A C9 D3 :F1
4580 7A FA 30 0D 5F F6 30 2D 7F CE 08 73 7C DE 38 FF :BC
4590 5A 3E 31 FF 66 00 03 FF 7E 00 03 FF 7F E0 00 FF :0E
45A0 7F FC 00 3F 7F FF FF FF 7F FF FF FF 6F FF FF FF :1E
45B0 67 FF FF FF 67 FF FF FF 67 FF FF FF 47 FF FF 7F :F0
45C0 47 FF FF 3F 47 FF FF 3F 67 FF FF 3F 67 FF FF 27 :38
45D0 61 FF F7 03 70 FF F3 03 70 FF F3 27 71 FF E3 3F :DA
45E0 43 FF C3 3F 41 FF C3 3F 41 FF E3 27 41 FF F3 03 :06
45F0 70 FF F3 03 70 7F F3 27 70 7F F3 3F 40 0F F3 3F :10
-----------------------------------------------------------
Sum: 7E DC ED E4 6E EC 6C AC 5F D3 C8 A0 56 AF CB B0 :B7


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4600 40 07 F3 3F 40 07 F3 3F 40 04 F2 3F 40 08 70 7F :9E
4610 60 38 70 FF 70 7C F1 FF 7C FF E3 F1 78 7F C1 C1 :AB
4620 70 3F C1 C1 70 3F 00 19 40 0E 00 1F 40 06 00 1F :CB
4630 40 02 00 13 40 02 00 01 40 06 00 21 40 0E 08 73 :C8
4640 40 1E 38 FF 42 3E 31 FF 66 00 03 FF 7E 00 03 FF :2D
4650 7F E0 00 FF 7F FC 00 3F 7F FF FF FF 7F FF FF FF :10
4660 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FE FF :EF
4670 7F FF FC 7F 7F FF FC 7F 7F FF F8 7F 7F FF F8 3F :9C
4680 7F FF E8 2F 7F FF F0 1F 7F FF F0 1F 7F FF D0 17 :14
4690 7F FF E0 0F 7F FF E0 0F 7F FF A0 0B 7F FF C0 07 :48
46A0 7F FF C0 07 7F FF C0 07 77 FF C0 07 63 FF 40 05 :6E
46B0 63 FF 80 03 63 FF 80 03 43 FF 80 03 42 FE 80 03 :52
46C0 41 FF 00 01 41 7F 00 01 40 FF 00 01 40 FF 00 01 :82
46D0 40 BF 00 01 40 6F 00 01 40 57 00 01 40 57 00 01 :E0
46E0 40 04 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :08
46F0 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
-----------------------------------------------------------
Sum: EE 3A 5F DA 00 E6 20 50 D7 66 9E 24 D6 E9 81 38 :2E


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4700 40 00 00 01 40 00 00 01 40 00 00 01 7F FF FF FF :3F
4710 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
4720 7F FF FE FF 7F FF FD 7F 7F FF FD 7F 7F FF F8 7F :64
4730 7F FF F8 3F 7F FF E8 2F 7F FF F0 1F 7F FF F0 1F :64
4740 7F FF D0 17 7F FF E0 0F 7F FF E0 0F 7F FF A0 0B :68
4750 7F FF C0 07 7F FF C0 07 7F FF C0 07 77 FF C0 07 :0C
4760 6B FF 40 05 6B FF 80 03 63 FF 80 03 43 FF 80 03 :46
4770 42 FE 80 03 41 FF 00 01 41 7F 00 01 40 FF 00 01 :05
4780 40 FF 00 01 40 BF 00 01 40 6F 00 01 40 57 00 01 :88
4790 40 57 00 01 40 04 00 01 40 00 00 01 40 00 00 01 :5F
47A0 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
47B0 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
47C0 FF FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :70
47D0 7F FF FF FF 7F F9 FF FF 7F F6 FC 7F 7F 8F 38 FD :25
47E0 7F B9 C9 F9 7F B3 B7 F9 7F DC 78 F1 7F ED CE E1 :BB
47F0 7F DD 86 C3 7F EA CD 83 7F F7 7A 07 7F FE FD 0F :DE
-----------------------------------------------------------
Sum: E4 E2 92 22 63 51 86 46 5B B0 F9 32 71 C8 C8 A2 :D3


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4800 73 FF 4A 0F 71 F8 B6 07 79 C9 FF 07 7D B7 FF 03 :6F
4810 7C 78 F3 83 7D CE E3 83 7D 86 27 83 7E CD DF 83 :75
4820 7F 7B E3 83 7E FD 3B 83 79 4B 1B 83 71 B7 37 03 :5D
4830 63 FD E9 07 7F 7B F4 07 7E 3D 28 07 7F 3E DB 03 :C7
4840 79 A7 F8 01 76 DB FC 01 4E 3C 7E 01 7A E7 5E 01 :30
4850 72 C3 63 01 5B 66 BB 01 6F BD 9A 01 5F 7E B4 01 :6F
4860 6A A5 EF 01 75 D5 F7 C1 7F 3E AF E1 7E 7F 5F E1 :91
4870 7E 7F FF E1 7F FF FF FF 7F FF FF FF 7F FF FF FF :51
4880 7F FF FF FF 7F FF FF FF 7F F9 FF FF 7F F0 FC 7F :58
4890 7F 80 38 FD 7F 80 09 F9 7F 80 07 F9 7F C0 00 F1 :64
48A0 7F E0 00 E1 7F C0 00 C3 7F E0 01 83 7F F7 02 07 :A4
48B0 7F FE 01 0F 73 FF 02 0F 71 F8 B6 07 79 C9 FF 07 :7E
48C0 7D 87 FF 03 7C 00 F3 83 7C 00 E3 83 7C 00 27 83 :00
48D0 7E 00 1F 83 7F 00 03 83 7E 00 03 83 79 00 03 83 :28
48E0 71 B0 07 03 63 FC 09 07 7F 78 04 07 7E 3C 08 07 :65
48F0 7F 3E D8 03 79 A7 F8 01 70 C3 FC 01 40 00 7E 01 :A0
-----------------------------------------------------------
Sum: 8B 4F 87 78 77 3A 76 AE 7F 99 D2 86 6A 08 0A FA :94


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4900 40 00 1E 01 40 00 03 01 40 00 03 01 60 00 02 01 :4A
4910 40 00 04 01 60 00 0F 01 75 D8 07 C1 7F 3E 0F E1 :77
4920 7E 7F 5F E1 7E 7F FF E1 7F FF FF FF 7F FF FF FF :12
4930 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
4940 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
4950 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
4960 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
4970 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
4980 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
4990 7F C0 0F FF 7C 00 00 FF 60 00 00 0F 40 00 00 03 :7A
49A0 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
49B0 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
49C0 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
49D0 40 00 00 01 40 00 00 01 40 00 00 01 7F FF FF FF :3F
49E0 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
49F0 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
```

ダンプ・リスト

```
Sum: 75 37 88 DE 92 77 09 DE 8C CF 01 CC D5 34 07 DE :18

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4A00 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
4A10 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
4A20 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
4A30 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
4A40 7F FF FF FF 7F C0 0F FF 7C 00 00 FF 60 00 00 0F :B3
4A50 40 00 00 03 40 00 00 01 40 00 00 01 40 00 00 01 :06
4A60 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
4A70 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
4A80 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
4A90 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
4AA0 7F FF 5F FF 7F FF BF FF 7F FF BF FF 7F FE AF FF :7F
4AB0 7F FF 1F FF 7F FF BF FF 7F FF BF FF 7F FF 2F FF :C0
4AC0 7F FF BF FF 7F FF FF FF 7F 5F FF FF 7C FF FF FF :0D
4AD0 7F FF FF FF 7F FF FF FF 7F FF B1 FF 7E 0F A0 7F :D2
4AE0 7C 1F A0 1F 70 07 A0 0F 60 07 80 07 60 03 00 07 :D8
4AF0 40 00 00 03 40 00 00 03 40 00 00 01 40 00 00 01 :08
-----------------------------------------------------
Sum: B2 15 D6 1E A6 BE 26 0C 93 5E A9 02 73 09 78 92 :73

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4B00 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
4B10 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
4B20 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
4B30 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
4B40 40 00 00 01 7F FF FF FF 7F FC E7 FF 7F FC 63 FF :FB
4B50 7F FE 67 FF 7F F7 5F FF 7F E3 B9 FF 7F F1 B1 FF :F1
4B60 7F FE A3 FF 7F FF 1F FF 7F F1 B9 FF 7E 70 B3 FF :83
4B70 7C 7F 2F FF 7E 67 9F FF 67 4F 81 FF 63 5C 03 FF :A3
4B80 73 BF 07 FF 7F 07 83 FF 78 0F 13 FF 7C 1D 31 FF :A2
4B90 7C 08 A0 7F 7C 19 A0 1F 70 07 A0 0F 60 07 80 07 :0B
4BA0 60 03 00 07 40 00 00 03 40 00 00 03 40 00 00 01 :31
4BB0 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 0F :12
4BC0 70 00 00 7F 7E 00 00 FF 7F E0 00 FF 7F F0 01 FF :39
4BD0 7F FC 03 EF 78 FF C7 83 7F F8 FF EF 70 37 7F C3 :7C
4BE0 7C E9 BF 9B 70 2C BE 43 74 A9 BE DB 70 37 7F DB :13
4BF0 77 B8 FF C3 74 BF FF DB FA F5 BE EF 74 EB 7D DD :53
-----------------------------------------------------
Sum: 2B E2 A1 54 50 66 C3 C2 B8 AB 08 CA 0E 26 F7 90 :2D

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4C00 6A D6 FB 9B 55 AD F7 37 6D 5B EE 6F 5A B7 DD 5F :78
4C10 79 61 8A FD 72 FB 25 FB 75 F6 A1 E7 40 00 9D DF :9D
4C20 40 00 19 9D 40 00 15 91 40 00 0C 41 40 00 0E 81 :38
4C30 40 00 0D 81 40 00 0B 01 40 00 07 01 40 00 08 01 :AB
4C40 40 00 0D 01 40 00 0B 01 40 00 06 01 40 00 0D 01 :2F
4C50 40 00 0E 01 40 00 0D 01 40 00 0B 01 40 00 07 01 :31
4C60 40 00 00 01 40 00 0D 01 40 00 0B 01 40 00 06 01 :22
4C70 40 00 0D 01 40 00 0B 01 40 00 03 01 40 00 0C 01 :2B
4C80 40 00 03 01 40 00 0F 01 7B 70 3F 01 42 0F C0 81 :51
4C90 76 F8 FD 8D 47 47 3B 33 7B 39 C4 0F 43 C6 78 7F :7B
4CA0 7A 3F 87 FF 40 00 00 1F 7F FF FF FF 40 00 00 01 :5B
4CB0 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
4CC0 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
4CD0 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
4CE0 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
4CF0 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
-----------------------------------------------------
Sum: D3 6E 5A 4C 4E EF B6 20 17 F9 C3 B0 1F 8C EE CA :E0
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4D00 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
4D10 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
4D20 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
4D30 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
4D40 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
4D50 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
4D60 43 3E 7F FF 66 0F 32 F3 7F 1B 40 61 7F F7 68 93 :45
4D70 7C E9 AF 9F 78 60 36 6F 7C 98 B6 23 7F BE B4 63 :71
```

```
4D80 73 A7 8B F7 61 03 83 FF 72 43 8D CF 7E FF BD 07 :D4
4D90 7E FF B2 CF 7D FF A0 FF 64 3F BF 7F 41 0F 3F 7F :08
4DA0 63 9F 9F BF 7B DF BF BF 71 BF BF BF 71 BF BF DF :B4
4DB0 7B BF BF DF 7F 7F 3F DF 7F 7F BF E7 7F 7F 9F CF :04
4DC0 7E FF BF EF 7C FF BF FF 7E FF BF FF 7D FF 3F FF :59
4DD0 7C FF 9F FF 79 FF BF FF 7D FF BF FF 7D FF BF FF :C3
4DE0 7F FF 3F FF 7F FF 9F FF 7F FF BF FF 7F FF FF FF :90
4DF0 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
-----------------------------------------------------
Sum: 06 27 65 F4 29 CB A5 00 3A 6F FC 7A 25 FD 72 2C :FE

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4E00 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
4E10 7F FF FF FF 43 3E 7F FF 66 0F 32 F3 7F 1B 40 61 :50
4E20 7F F7 68 93 7C E9 AF 9F 78 60 36 6F 7C 98 B6 23 :8E
4E30 7F BE B4 63 73 A5 8B F7 61 01 83 FF 72 43 8D CF :E3
4E40 7E FB BD 07 7E F7 B2 CF 7D F1 A0 FF 64 30 BF 7F :12
4E50 41 09 27 63 63 9F 83 A1 7B DF A7 83 71 BF BC BF :29
4E60 71 B9 B8 5F 7B B0 BC DF 7F 79 33 D3 7F 7F A1 21 :C5
4E70 7F 7F 92 09 72 FF BF 2F 60 F9 BF E7 72 F0 B3 C7 :D3
4E80 7D 39 21 C7 64 1F 93 EF 41 3F BF FF 75 F9 BF FF :0D
4E90 7D F0 B3 FF 78 F9 21 FF 78 FF 93 FF 7D FF BF FF :F3
4EA0 7F FF 9F FF 7F FF 8F FF 7F FF 8F FF 7F FF DF FF :90
4EB0 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
4EC0 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
4ED0 7F FF FF FF 7F FF FF FF 7F FF FF F9 7F FF FF F1 :DC
4EE0 7F FF FF E3 7F FF FF E7 7F FF FF C7 7F FF FF C7 :4C
4EF0 7F FF FF 8F 5F FF CF 8F 47 FB FF 8F 61 FF FF 1F :16
-----------------------------------------------------
Sum: 9F 12 B6 FA 35 22 76 72 10 E5 FF E6 80 45 A9 4A :32

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4F00 70 FF FF 1F 78 7F FF 1F 7C 2F FC 9F 7F FF F9 7F :DE
4F10 7F 1F FB B9 7E 4D E7 B1 7E E0 0F B3 7E FB F7 A3 :E8
4F20 7C F7 FB A3 7D EF FD 27 7D E3 F8 67 7C 1D FB C7 :BB
4F30 7B F6 E6 CF 7F 07 5C 0F 71 03 B8 0F 60 03 F0 1F :C4
4F40 60 03 70 1F 60 03 70 11 60 03 20 01 60 01 20 07 :E2
4F50 66 01 A0 1F 66 01 A0 31 77 01 E0 63 47 00 C0 47 :67
4F60 63 00 C0 0F 63 00 C0 0F 61 80 C0 1F 70 00 40 1F :F3
4F70 70 00 40 3F 78 00 00 3F 78 00 20 3F 7F FF FF FF :F9
4F80 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF F9 :EA
4F90 7F FF FF F1 7F FF FF E3 7F FF FF E7 7F FF FF C7 :76
4FA0 7F FF FF C7 7F FF FF 8F 5F FF CF 8F 47 FB CF 8F :AC
4FB0 61 F3 E7 1F 70 E6 77 1F 78 66 7F 1F 7C 2C FC 9F :05
4FC0 7F FC F8 7F 7F 1C F8 39 7E 0D E0 31 7E 00 00 33 :0B
4FD0 7E 00 00 23 7C 00 00 23 7C 00 00 27 7C 00 00 67 :C6
4FE0 7C 1C 03 C7 7B F6 06 CF 7F 07 1C 0F 71 03 B8 0F :94
4FF0 60 03 F0 1F 60 03 F0 1F 60 03 F0 11 60 03 E0 01 :8C
-----------------------------------------------------
Sum: 36 1A BA 34 56 BE 71 70 46 F3 D3 96 FB 45 5B 0C :7C

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5000 60 01 E0 07 66 01 E0 1F 66 01 E0 31 77 01 E0 63 :E1
5010 47 00 C0 47 63 00 C0 0F 63 00 C0 0F 61 80 C0 1F :72
5020 70 00 40 1F 70 00 40 3F 78 00 00 3F 78 00 20 3F :4C
5030 FF FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :70
5040 7F FF FF FF 7F F7 FF FF 7F F7 FF FF 7F E3 FF FF :C4
5050 7F F3 FF FF 7F FB FF FF 7F EB FF FF 7F F3 FF FF :C0
5060 7F FC 7F FF 7F FD FF 7F 7F FD FF 3F 7F F9 FF 7F :A3
5070 7F F3 FC 7F 7F F3 FE FF 7F F7 EE FF 7D F7 E4 7F :96
5080 78 B7 C5 FF 7F 66 68 FF 7F AC 30 3F 4E 4E 34 7F :28
5090 44 5F C7 FF 64 D9 E8 FF 55 90 C0 7F 4B B8 DC FF :8F
50A0 43 B8 DF FF 63 3A 17 FF 7B 7E 61 FF 4C 7C F1 FF :9D
50B0 46 F1 FD FF 64 C7 FF FF 59 9B FF FF 48 31 FF FF :C5
50C0 4B F3 FF 3F 73 E6 00 7F 72 00 FF 7F 60 6E FF FF :10
50D0 66 E6 7F FF 4C 66 3F FF 5E 23 3F FF 7F 7B FF FF :71
50E0 7F FF FF FF 7F FF FF FF 7F F7 FF FF 7F C7 FF FF :B0
50F0 7F C3 FF FF 7F E3 FF FF 7F 37 FF FF 7E 16 7F FF :66
-----------------------------------------------------
Sum: 06 3B 3C 20 1B 50 7D 60 32 7C 16 F2 D2 BF 1C 34 :7C
```

グラフィック花札こいこい

ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5100 7F 20 3F FF 7F F2 7F FF 7F 3B FF FF 7E 0B FF 7F :8B
5110 7F 33 9E 3F 7F FC 0E 3F 7F 9D 9F 67 7F 8D F3 03 :7B
5120 7F C9 E1 67 67 F3 F0 7F 6B B2 7E FF 73 14 2E 9F :47
5130 65 12 64 0F 40 B7 C5 1F 67 66 68 FF 7F AC 30 3F :93
5140 4E 4E 34 7F 44 5F C7 FF 64 D9 E8 FF 55 90 C0 7F :00
5150 4B B8 DC FF 43 A0 DF FF 63 02 17 FF 7B 66 01 FF :FB
5160 4C 7C F1 FF 46 F1 FD 7F 64 C7 EE 3F 59 9B C6 3F :BC
5170 48 31 C7 67 4B F3 EF 03 73 E6 00 67 72 00 F7 7F :7F
5180 60 6E E3 7F 66 E6 62 3F 4C 66 36 3F 5E 23 3F 7F :83
5190 7F 7B FF FF 7F FF FF FF FF FF FF FF 7F FF FF FF :EC
51A0 7F E0 FF FF 7F F0 13 8F 7F FE 65 03 7F FF 6C 3F :7C
51B0 7F FF 6C FF 7F 80 6D FF 7F 01 6D FF 7F FE 01 FF :BD
51C0 7F FF C0 FF 7D F7 E0 3F 7E 23 E2 1F 7E 03 C7 FF :B9
51D0 7E 01 8F FF 7E 01 1F DF 78 01 3F 9F 78 01 3C 1F :B5
51E0 7C 00 38 1F 7E 00 38 0F 7E CF 30 01 7E CF 90 01 :F4
51F0 7C EF A0 01 7D F1 30 3F 60 EE 2B FF 6B 0D D9 FF :B1
-------------------------------------------------------
Sum: E1 98 5E 32 96 B9 1C 94 8B BD F4 06 44 E8 E5 76 :D1

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5200 5A 51 CD FF 5A AA 0D FF 5B 14 6D FF 5C E1 EC 07 :92
5210 69 FD C8 01 76 70 20 01 77 06 F0 01 77 BE 70 01 :4A
5220 73 3E 60 01 7C 1D 40 01 78 01 80 01 78 07 F0 01 :56
5230 70 07 F8 01 60 07 F9 01 78 0F FD C1 71 EF FF F9 :6E
5240 7B FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :EC
5250 7F FF FF FF 7F E0 FF FF 7F F0 13 8F 7F FE 01 03 :6B
5260 7F FF 00 3F 7F FF 00 FF 7F 80 01 FF 7F 01 01 FF :B9
5270 7F FE 01 FF 7F FF C0 FF 7D F7 E0 3F 7E 23 E2 1F :EF
5280 7E 03 C7 FF 7E 01 8F FF 7E 01 1F DF 78 01 3F 9F :28
5290 78 01 3C 1F 7C 00 38 1F 7E 00 38 0F 7E 00 30 01 :1B
52A0 7E 00 10 01 7C 00 00 01 7C 00 00 3F 60 00 03 FF :29
52B0 60 00 01 FF 40 00 01 FF 40 00 01 FF 40 00 01 FF :20
52C0 40 00 00 07 60 00 00 01 70 00 00 01 70 00 00 01 :8A
52D0 70 00 00 01 70 00 00 01 7C 01 00 01 78 01 80 01 :5A
52E0 78 07 F0 01 70 07 F8 01 60 07 F9 01 78 0F FD C1 :86
52F0 71 EF FF F9 7B FF FF FF 7F FF FF FF 7F FF FF FF :C8
-------------------------------------------------------
Sum: 0B 88 EF 5E 19 22 E3 1E 3F 98 1D BC 2C C6 1D 82 :5D

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5300 00 00 00 00 80 00 00 00 00 00 0E 00 00 00 05 88 :1B
5310 00 00 06 B0 00 00 1C 60 00 00 7F B0 00 00 01 DB :3A
5320 00 00 00 FC 00 00 01 CE 00 00 00 06 00 00 00 00 :D1
5330 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5340 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5350 00 00 00 00 00 00 00 00 04 00 00 00 0C 00 00 00 :10
5360 0C 00 00 00 1E 00 00 00 3A 00 18 00 3A 00 3E 00 :F4
5370 33 F8 0C 00 28 F0 00 18 3E F0 00 F0 33 B8 01 B0 :21
5380 25 9C 03 58 0F 50 07 B8 1E 5E 0C 18 3E D8 00 18 :08
5390 0C C3 00 08 31 FE 00 08 3F 1C 00 00 1C 70 00 00 :F5
53A0 0F A0 00 00 0F D0 00 00 3C FC 00 00 38 3E 00 00 :3C
53B0 30 0F 00 00 7F FF FF FF 7F FF C1 FF 7F FF E0 61 :B8
53C0 7E FF F0 03 7F 3D C0 03 7F 0C 80 07 7F 84 40 03 :47
53D0 7F 80 00 01 7F 80 02 01 7E 00 04 01 70 00 08 01 :FE
53E0 60 00 07 F9 7F E0 07 FF 7F C0 87 FF 7F C3 C3 FF :8E
53F0 7F CF F3 FF 7F DF FB FF 7F FF FF FF 7F FF FF 7F :10
-------------------------------------------------------
Sum: 8B 54 FF 08 90 89 E7 07 8F 30 7C C3 77 83 2F 0B :1F

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5400 7F FF FF 1F 7B FF FF 87 73 FF FF 83 65 FF FF C1 :B4
5410 6D FF FE 01 6C FF C0 01 5E FF E0 01 7A FF C0 01 :0F
5420 7A 01 80 01 73 FB C0 01 68 F7 F0 03 7E F7 F0 03 :E5
5430 73 BB F8 03 60 9D F8 03 43 10 F0 03 46 1F E0 43 :EF
5440 46 18 23 C3 4C 00 3F E3 40 00 7F E3 40 00 FF F7 :8A
5450 40 01 FF FF 60 0F FF FF 40 01 FF FF 40 00 FF FF :29
5460 41 00 7F FF 47 C0 7F FF 00 00 00 00 80 00 00 00 :C4
5470 00 01 00 00 00 20 80 00 00 17 C4 00 00 0F EE 00 :79
5480 00 1F FC 00 00 3F F8 00 00 1C 3C 00 00 00 00 00 :AA
5490 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
54A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
54B0 00 00 00 00 00 00 3F 80 00 00 7F C0 00 00 7F C0 :3D
54C0 00 07 FF C0 00 0F FF C0 00 0F FF E0 00 3F FF E0 :A0
54D0 00 7F FF E0 00 7F FF C0 00 3F FF C0 00 1F FF 00 :B8
54E0 00 0F FF 00 00 06 07 00 00 00 00 00 00 00 00 00 :1B
54F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
-------------------------------------------------------
Sum: A0 88 0F 85 AD 58 F0 6D FC 87 BA CC A3 81 F8 9E :E1

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5500 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5510 00 00 00 00 00 00 00 00 00 00 00 00 7F FF FF FF :7C
5520 7F FF FF FF 7F FE FF FF 7F DF 7F FF 7F EE BB FF :FA
5530 7F F2 97 FF 7F E2 DF FF 7F C2 D7 FF 7F E8 1B FF :DE
5540 7F 9C 3F FF 78 0C 3F FF 78 06 7F FF 7C 00 E1 FF :73
5550 7E 00 40 1F 7F 86 00 0F 7E FE 40 07 6C 7E 60 0F :0D
5560 44 0F 3C 1F 40 0F 3F FF 40 0F 97 79 40 07 B2 71 :04
5570 40 0F F2 41 40 0B 34 C1 70 00 BD 81 78 7B CA 21 :4E
5580 7F FC FE 21 71 C1 09 A1 70 C2 4C C1 60 24 C8 01 :02
5590 40 19 4D 61 40 0E EE 73 40 07 E5 33 70 07 C0 33 :7F
55A0 70 0F 00 33 79 0E 00 73 7F 0C 00 71 7F 9C 00 21 :E4
55B0 7F FC 00 01 7F FE 00 01 7F FF 81 03 7F FF C3 87 :C4
55C0 7F FF E7 87 7F FF FF 8F 7F FF FF 1F 7F FF FE 3F :4F
55D0 FF FF FF FF 7F FF FF FF 7F FF FF FF 7F FF F7 FF :68
55E0 7F FF F3 FF 7F FF 93 FF 7B FF 63 FF 79 FF 63 FF :36
55F0 79 FF 92 7F 79 FF F1 BF 79 FF F1 BF 79 3F F2 7F :02
-------------------------------------------------------
Sum: A3 C7 F9 36 14 63 09 A0 44 84 6D 42 DB D7 27 35 :3E

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5600 78 DF F3 FF 78 DF 33 FF 61 3E D3 FF 59 FE D3 FF :6C
5610 59 FF 33 FF 61 FF F3 FF 79 3F F3 FF 78 DF F3 FF :CF
5620 78 DC 43 FF 79 3B BD FF 78 FB FD FF 78 FD FD FF :E6
5630 7C 7E 7B FF 7F 7D BD FF 7C FB DE FF 7          FF :23
5640 77 BF F9 FF 67 8E FB 9F 5E F4 77 6F 5C 71 23 6D :52
5650 69 26 C0 99 76 C6 C0 31 76 C0 38 03 79 00 FE 07 :04
5660 7F 01 3F 8F 7E 02 DF FF 7C 02 DF FF 78 07 3F FF :C5
5670 70 0F FF FF 60 3F FF FF 40 3F FF FF 40 3F FF FF :14
5680 40 7F FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :31
5690 7F FF F7 FF 7F FF F3 FF 7F FF 93 FF 7B FF 03 FF :70
56A0 79 FF 03 FF 79 FF 92 7F 79 FF F0 3F 79 FF F0 3F :51
56B0 79 3F F2 7F 78 1F F3 FF 78 1F 33 FF 61 3E 13 FF :2C
56C0 41 FE 13 FF 41 FF 33 FF 61 FF F3 FF 79 3F F3 FF :BF
56D0 78 1F F3 FF 78 1C 43 FF 79 38 01 FF 78 F8 01 FF :80
56E0 78 FC 01 FF 7C 7E 03 FF 7F 7C 01 FF 7C F8 00 FF :DE
56F0 78 60 00 FF 70 00 01 FF 60 00 03 9F 40 00 07 0F :9F
-------------------------------------------------------
Sum: EE 62 CD 9A 20 E0 2A 42 06 37 DB 44 D2 5E E8 B6 :4D

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5700 40 00 03 0D 60 00 00 99 70 00 00 31 70 00 38 03 :95
5710 79 00 FE 07 7F 01 3F 8F 7E 02 1F FF 7C 02 1F FF :06
5720 78 07 3F FF 70 0F FF FF 60 3F FF FF 40 3F FF FF :54
5730 40 3F FF FF 40 7F FF FF 7F FF FF FF 7F FF FF FF :32
5740 7F FF FF FF 7F FF FF FF 7F FF FF FF 6F FF FF FF :E0
5750 47 FF FF FF 47 FF FF FF 47 FF FF FF 42 FF FF FF :0B
5760 41 FF FF FB 41 7F FF F1 40 FF FF F1 40 FF FF F1 :48
5770 40 BF FF A1 40 7F FF C1 40 7F FF 41 40 5F FF 81 :3C
5780 40 3F FF 81 40 3F FE 81 40 3D FF 01 40 28 FF 01 :E2
5790 40 18 BD 01 40 10 7E 01 40 00 7E 01 40 00 7E 01 :63
57A0 40 00 7A 01 40 00 7C 01 40 00 7C 01 40 00 1C 01 :92
57B0 40 00 04 01 40 00 00 01 40 00 00 01 40 00 00 01 :08
57C0 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
57D0 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
57E0 40 00 00 01 40 00 00 01 40 00 00 01 7F FF FF FF :3F
57F0 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
-------------------------------------------------------
Sum: F7 58 74 32 15 D9 30 5C 52 F8 11 64 5A C2 E8 74 :A6

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
```

```
5800 6F FF FF FF 57 FF FF FF 57 FF FF FF 47 FF FF FF :58
5810 42 FF FF FF 41 FF FF FB 41 7F FF F5 40 FF FF F5 :60
5820 40 FF FF F1 40 BF FF A1 40 7F FF C1 40 7F FF 41 :4C
5830 40 5F FF 81 40 3F FF 81 40 3F FE 81 40 3D FF 01 :99
5840 40 2A FF 01 40 18 BD 01 40 10 7E 01 40 00 7E 01 :0E
5850 40 00 7E 01 40 00 7A 01 40 00 7C 01 40 00 7C 01 :F4
5860 40 00 1C 01 40 00 04 01 40 00 00 01 40 00 00 01 :24
5870 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
5880 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
5890 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
58A0 FF FF FF FF 7F FF E7 FD 7F CF F3 F9 7C B7 F3 F9 :B7
58B0 7B 78 FB 9B 47 8E FF 6F 5C EE F8 F1 58 6D FB 9D :5C
58C0 6C FB FB 0D 77 CD FD DB 6F B7 FE 77 74 78 FD BB :CA
58D0 7D CE E3 C7 7D 86 EE 77 7E CD EC 37 7F 7B F6 6F :2A
58E0 7A FD FB DF 73 4B F7 EF 73 B6 FA 5F 63 FC 45 BF :DA
58F0 67 B0 07 FF 7F 98 0F FF 7F 8C 1F FF 7F CC 0F FF :C4
-------------------------------------------------------
Sum: F5 73 6F C2 44 D7 0E CE 52 CF E3 32 30 99 2B BA :74

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5900 7F FE 07 FF 7F FE 07 9F 7C FF 07 1F 7C 7C C7 33 :39
5910 7E 7B 7F 63 7F 47 8F E7 7F DC EE 6F 7F D8 6D BF :52
5920 7F EC C3 C7 7F F7 AE 77 7F AF CC 37 7F 34 B6 6F :99
5930 7C 2B 7B DF 78 2C 37 EF 60 36 6A 5F 40 2B DD BF :31
5940 40 17 EF DF 40 0A 5F CF 40 1D BF C7 40 3F FF CF :CD
5950 40 7F FF FF 7F FF FF FF 7F FF E7 FD 7F CF F3 F9 :D5
5960 7C 87 F3 F9 78 00 FB 9B 40 00 FF 0F 40 00 F8 01 :84
5970 40 01 F8 01 60 03 F8 01 70 01 FC 03 60 07 FE 07 :72
5980 74 00 FC 03 7C 00 E0 07 7C 00 E0 07 7E 01 E0 07 :9F
5990 7F 03 F0 0F 7A 01 F8 1F 73 03 F0 0F 73 B6 F8 1F :C8
59A0 63 FC 45 BF 67 B0 07 FF 7F 98 0F FF 7F 8C 1F FF :CE
59B0 7F CC 0F FF 7F FE 07 FF 7F FE 07 9F 7C FF 07 1F :A0
59C0 7C 7C C7 33 7E 78 7F 63 7F 40 0F E7 7F C0 0E 6F :3B
59D0 7F C0 0D BF 7F E0 00 07 7F F0 00 07 7F A0 00 07 :0D
59E0 7F 30 00 0F 7C 20 00 1F 78 20 00 0F 60 30 08 1F :D7
59F0 40 28 1D BF 40 10 0F DF 40 08 1F CF 40 1D BF C7 :9B
-------------------------------------------------------
Sum: C3 0D CE 70 21 AB 40 E2 EC CE E0 7A A3 B7 82 90 :7C

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5A00 40 3F FF CF 40 7F FF FF 7F FF FF FF 7F FF FF FF :02
5A10 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
5A20 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
5A30 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
5A40 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
5A50 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
5A60 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
5A70 7F C0 0F FF 7C 00 00 FF 60 00 00 0F 40 00 00 03 :7A
5A80 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
5A90 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
5AA0 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
5AB0 40 00 00 01 40 00 00 01 40 00 00 01 7F FF FF FF :3F
5AC0 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
5AD0 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
5AE0 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
5AF0 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
-------------------------------------------------------
Sum: B5 F5 04 C8 B2 75 F5 F8 D5 F5 F5 08 F4 F4 F4 FA :27

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5B00 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
5B10 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
5B20 7F FF FF FF 7F C0 0F FF 7C 00 00 FF 60 00 00 0F :B3
5B30 40 00 00 03 40 00 00 01 40 00 00 01 40 00 00 01 :06
5B40 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
5B50 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
5B60 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
5B70 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF 7F FF :70
5B80 7F FF 7F FF 7F FF 7F FF 7F FF 3F FF 7D FE 7F FF :AD
5B90 7D FF 7F FF 7C FF 7F FF 7D FD 3F FF 7E FE 7F 7F :25
5BA0 7F FE 7F 7F 7F FF 7F 7F 7F FF FF 7F 7C FF FE 7F :EB
5BB0 78 7F FF 3F 78 0F FF 7F 70 03 FC 7F 60 01 F8 7F :00
5BC0 60 00 30 0F 40 00 30 07 40 00 00 07 40 00 00 07 :A4
5BD0 40 00 00 07 60 00 00 03 70 00 00 03 78 00 00 03 :98
5BE0 78 00 00 03 78 00 00 03 7C 00 20 07 7E 01 FE 07 :1D
5BF0 7E 07 FF C7 7F 1F FF FF 7F 7F FF FF 7F FF FF FF :5F
-------------------------------------------------------
Sum: 85 7E A7 9E 85 E8 B7 08 8F 7A 95 0C 69 F9 6E 9C :8A

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5C00 7F FF FF FF 7B FB 6F DF 70 71 AF 8F 7B FA EF DF :A2
5C10 7A 3A EF 8F 7B FA EF DF 7B FA ED 87 7A 3E F3 97 :A0
5C20 7F FF FF FF 7F FF FF FF 7F FC FF FF 7F FC 67 FF :52
5C30 7F F2 67 FF 73 F1 4F FF 71 99 5F FF 79 1D 27 FF :AD
5C40 6D 3A 67 9F 65 71 6F 9F 64 F9 5F 1F 71 FD 3F 3F :58
5C50 7E FE 79 7F 7C 1E 79 67 60 3C 79 4F 70 60 0D 4F :7E
5C60 78 30 1E 5F 78 20 3F 3F 78 02 3F 7F 70 03 BC 7F :21
5C70 60 01 F8 7F 60 00 30 0F 40 00 30 07 40 00 00 07 :35
5C80 40 00 00 07 40 00 00 07 60 00 00 03 70 00 00 03 :64
5C90 78 00 00 03 78 00 00 03 78 00 00 03 7C 00 20 07 :14
5CA0 7E 01 FE 07 7E 07 FF C7 7F 1F FF FF 7F 7F FF FF :67
5CB0 7F FF FF FF 7F FF FF FF 7B FB 6F DF 70 71 AF 8F :DB
5CC0 7B FA EF DF 7A 3A EF 8F 7B FA EF DF 7B FA ED 87 :A1
5CD0 7A 3E F3 97 7F FF FF FF FF C0 FF FF 4F 80 78 0F :D1
5CE0 43 80 70 07 41 80 20 03 40 80 20 03 40 C0 20 01 :22
5CF0 40 40 20 01 40 40 70 01 40 60 70 01 40 40 10 01 :34
-------------------------------------------------------
Sum: E7 8B B9 16 D0 93 7F 72 23 EB 2D CE A3 1B DB B8 :EF

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5D00 40 5F D0 01 40 5F D8 01 40 5F D8 01 40 6F E8 01 :F8
5D10 40 2F E8 01 40 2F E8 01 50 2F E8 01 50 37 F0 01 :90
5D20 50 37 F3 01 50 37 F7 01 50 3B FA 01 48 1B FA 01 :DE
5D30 48 1D FC C1 4C 1D FD C1 44 1E FE C3 46 1E FE E3 :B1
5D40 46 1F 7F 67 47 0F 7F 67 43 0F BF 8F 43 8F DF DF :B7
5D50 43 CF E0 0F 43 CF FF FF 41 EF FF FF 40 7F FF FF :FC
5D60 40 3F FF FF 40 3F FF FF 60 3F FF FF 60 1F FF FF :14
5D70 60 1F FF FF 70 0F FF FF 78 0F FF FF 7C 07 FF FF :00
5D80 7C 07 FF FF 7E 03 FF FF 7E 01 FF FF 7F C0 FF FF :BA
5D90 4F 80 78 0F 43 80 70 07 41 80 20 03 40 80 20 03 :57
5DA0 40 C0 20 01 40 40 20 01 40 40 70 01 40 60 70 01 :C4
5DB0 40 40 10 01 40 40 10 01 40 40 18 01 40 40 18 01 :54
5DC0 40 60 08 01 40 20 08 01 40 20 08 01 50 20 08 01 :F4
5DD0 50 30 00 01 50 30 03 01 50 30 07 01 50 38 02 01 :18
5DE0 48 18 02 01 48 1C 00 C1 4C 1C 01 C1 44 1E 00 C3 :D7
5DF0 46 1E 00 E3 46 1F 00 67 47 0F 00 67 43 0F 80 0F :B1
-------------------------------------------------------
Sum: AA 7B B5 2E B5 9C DA 5A E2 AF 2B 80 E3 78 DD 9A :9B

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5E00 43 8F C0 1F 43 CF E0 0F 43 CF FF FF 41 EF FF FF :F0
5E10 40 7F FF FF 40 3F FF FF 40 3F FF FF 60 3F FF FF :54
5E20 60 1F FF FF 60 1F FF FF 70 0F FF FF 78 0F FF FF :FC
5E30 7C 07 FF FF 7C 07 FF FF 7E 03 FF FF 7E 01 FF FF :FE
5E40 00 00 00 00 00 00 00 00 80 00 00 00 00 00 00 00 :80
5E50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5E60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5E70 00 7F 8E 00 00 7F 9F 00 00 7F 86 00 00 7F 80 0C :3B
5E80 00 3F 80 1E 00 3F C0 2C 00 3F C6 00 00 3F CF 00 :1B
5E90 00 3F C6 8C 30 1F C0 0E 3C 9F E0 1C 19 9F E0 00 :1D
5EA0 00 9F E6 00 18 1F E7 0C 3C 0F E3 9E 18 CF F0 0C :5E
5EB0 01 CF F0 40 00 CF F0 60 18 0F F8 E0 3C 07 F8 E0 :39
5EC0 18 67 F8 40 00 F7 FC 00 00 E3 FE 00 02 03 FE 00 :8E
5ED0 03 81 FC 00 03 81 F0 00 01 00 00 00 00 00 00 00 :F5
5EE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5EF0 00 00 00 00 73 BF 0C A3 60 B2 CF C1 73 61 7F D3 :A9
-------------------------------------------------------
Sum: 7B 87 5B 46 1D 36 CB 55 E2 30 D0 57 79 D5 90 C7 :F4

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5F00 7E F3 78 DF 66 9F B1 2F 41 0C C9 09 65 98 1D 21 :07
5F10 7D FE 6D E9 78 7F C6 EF 48 3F C6 C7 47 FF EF 47 :0D
```

グラフィック花札こいこい

ダンプ・リスト

```
5F20 47 00 3F 6F 77 00 3F 7F 73 00 3F BF 73 00 3F 3F :8C
5F30 63 00 3F BF 67 80 3F AF 6F 80 1F 9F 7F 80 1F BF :C0
5F40 7E 80 1F BF 7E 80 1F 3F 7D C0 1F DF 79 C0 0F CF :8A
5F50 7C C0 0F DF 7D C0 0F DF 7D C0 0F DF 79 E0 0F DF :C7
5F60 7D E0 07 BF 7C E0 07 BF 7D E0 07 BF 7D E0 03 F7 :BF
5F70 7A F0 03 FF 7E F0 03 FF 7E 70 01 FB 7E F8 00 FF :3B
5F80 7E F8 00 FF 7F FC 00 FB 7F FC 03 FF 7F FE 0F F7 :EB
5F90 7F FF 3F FF 7F FF FB FB 7F FF FF FF 7B BD F6 FF :D9
5FA0 6F F7 BF BF 7F FF FF FF 00 00 00 00 00 00 00 00 :60
5FB0 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :80
5FC0 00 7F 00 00 00 7F 00 00 00 7F 00 00 00 3F 80 00 :3C
5FD0 00 3F 80 00 00 3F 80 00 00 3F 80 00 00 3F 80 00 :FC
5FE0 00 1F C0 00 00 1F C0 00 02 1F C0 00 02 1F C0 00 :80
5FF0 00 0F E0 00 00 0F E0 00 00 0F E0 20 0F 07 F0 20 :13
-------------------------------------------------------
Sum: 82 DB B9 0F 2E 94 47 1D 60 82 45 C4 96 EE 40 20 :1A

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6000 1F 87 F0 20 3F 87 F0 00 3F 43 F8 00 3E E3 F8 00 :FF
6010 0D E3 F8 00 01 C1 FC 00 00 81 FC 00 00 00 FE 00 :21
6020 00 00 FE 00 00 00 00 00 00 00 00 00 00 00 00 00 :FE
6030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6050 00 00 00 00 00 00 00 00 00 00 00 00 7F FF FF FF :7C
6060 7F BF EF 7F 77 FD FF EF 7F FF FD FF 7F FF FF FF :04
6070 7F 00 7F BF 7F 00 7F FD 47 00 7F F1 63 00 7F E3 :34
6080 71 80 3F C7 78 80 3F 8F 78 80 3F 8F 7C 80 3F 1F :DD
6090 7F 80 3F 1F 7F C0 1E 3B 78 C0 1E 7F 7A C0 1C 7F :9F
60A0 7A C0 1C FF 78 60 08 DF 7F E0 08 8F 70 E0 08 AF :11
60B0 6F 70 05 AF 5F B0 05 AF 7F B0 07 8F 7F 48 03 B9 :9E
60C0 7E E8 03 71 4D E8 02 E7 71 DC 01 C7 7E AC 01 8F :C7
60D0 7F 6E 00 9F 4F C6 00 FF 43 85 00 7F 61 C5 FF F1 :FD
60E0 78 C0 33 E3 6C 60 23 87 66 20 63 0F 72 20 42 1F :AF
60F0 48 00 44 39 44 00 00 63 62 00 00 47 60 00 00 0F :84
-------------------------------------------------------
Sum: C0 6F 6D 1E 50 A3 F9 14 6F 14 40 B8 35 DA 1B 95 :F4

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6100 60 00 00 1F 70 00 00 1F 70 00 00 1F 78 00 00 3F :54
6110 FF FF FF FF 7F FF 7B FF 7F FD FF FF 7F FF DF FF :CA
6120 7D FF FF FF 7C FF 7F FF 7E FF FF 7F 7F 67 FF 7F :D2
6130 7F 40 03 7F 7F 2B FB 7F 7F BB FB 7F 7F DB FB 3F :AD
6140 6F DB FB 3F 7F AB FB 7F 7F FB F7 7F 6F F7 F7 7F :F4
6150 7F F7 F7 3F 7F F7 F7 7F 7F EF ED 7F 7F EF E8 73 :3B
6160 7F EF E9 63 7E 5F DF 67 76 DF DF 0D 78 5F D8 B9 :86
6170 79 BF B0 19 7E BF B8 C7 7F 7F 7F E7 7F 7F 7F 99 :37
6180 7E FE DF 1F 7E FE CE 3F 7D FD C6 7F 7D FD E7 FF :22
6190 7B FB F7 FF 70 03 F8 37 7F F9 83 87 7F F0 77 F3 :69
61A0 6F E1 E7 3F 7F FF C6 1F 77 FF C7 0F 7F BF CF F7 :29
61B0 7F FD FF FB 6E FF FF E5 7F EF FF C7 7F FE FF CF :46
61C0 7F FF FF FF 7D FF FF FF 78 FF 7B FF 78 FD FF 7F :DA
61D0 7D 9F DE 3F 65 0F FE 3F 40 9F 7F 67 66 FF F9 43 :50
61E0 7F 67 FE 67 79 40 03 7F 70 28 03 7F 79 B8 02 67 :3A
61F0 7F D8 03 03 6C D8 03 27 78 28 03 7F 7C F8 05 7F :E5
-------------------------------------------------------
Sum: 22 72 26 96 86 0E 0C 26 71 D1 4A 4E A7 5B 3A A0 :CC

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6200 6F F0 06 67 7F F0 05 03 7F F0 07 67 77 E0 0D 7F :03
6210 63 E0 08 73 63 E0 09 63 76 40 1F 67 76 C0 1F 0D :0B
6220 78 40 18 B9 49 80 30 19 46 80 38 C7 4F 00 7F E7 :15
6230 7F 00 7F 99 7E 00 DF 1F 7E 00 CE 3F 7C 01 C6 77 :58
6240 7C 01 E7 E3 78 03 F7 E3 70 03 F8 37 7F F9 83 87 :C0
6250 7F F0 77 F3 6F E1 E7 3F 7F FF C6 1F 77 FF C7 0F :FE
6260 7F BF CF F7 7F FD FF FB 6E FF FF E5 7F EF FF C7 :FF
6270 7F FE FF CF 7F FF FF FF 7F FF FF FF 7F FF E3 EF :93
6280 7C 7E 01 FF 7E 10 0F 7F 77 00 1F FB 7F E0 03 EF :F8
6290 FF 80 03 FF 6E 00 79 FF 7E 7C FF 7F 7F FC 7F 3F :18
62A0 77 FF FE 01 7F F0 06 03 7F F7 F4 07 6E F7 F4 0F :C6
62B0 7F F7 F4 3F 7F F7 F4 7F 7F EF EC 7F 79 EF E8 F1 :AC
62C0 60 6F E8 03 40 2F E8 03 40 1F D0 07 40 5F D0 03 :BC
62D0 60 5F D0 03 70 BF 80 07 61 BF 80 03 7F BF 80 01 :AA
62E0 7F 7F 00 01 77 7F 00 01 7E FE 00 01 6E FE 00 01 :E0
62F0 7D FC 00 01 7D FC 00 01 7B F8 00 01 7B F8 00 01 :DC
-------------------------------------------------------
Sum: EF FB 7F 0E 1C 90 E3 C6 22 E6 36 1A 39 5D 4B 6A :6F
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6300 77 F0 00 03 70 00 00 03 7F F8 00 01 6F C0 08 01 :8D
6310 7F 80 18 01 7F 00 18 01 7B C0 3C 03 77 CE 2C 03 :9E
6320 7F FF 7F C3 7E FF F7 FF 7F FF FF FF 7F FF FF FF :2B
6330 7F FF E3 EF 7C 7E 01 FF 7E 10 CF 7F 77 06 DF FB :7D
6340 7F EE C3 EF 7F 86 83 FF 6E 00 79 FF 7E 7C FF 7F :04
6350 7F FC 7F 3F 77 FF FE 01 7F F0 06 03 7F F0 04 07 :A0
6360 6E F0 04 0F 7F F0 04 3F 7F F0 04 7F 7F E0 0C 7F :FF
6370 79 E0 08 F1 60 60 08 03 40 20 08 03 40 00 10 07 :DF
6380 40 40 10 03 60 40 10 03 70 80 20 07 61 80 20 F3 :51
6390 7F 80 3E F9 7F 00 5C 0F 77 00 61 F7 7E 00 EE 41 :9C
63A0 6E 00 C1 B7 7C 01 85 B7 7C 01 C0 77 78 02 FB B7 :7F
63B0 78 03 38 2D 70 07 C7 DB 70 05 EF E3 7F F8 77 81 :AF
63C0 6F C0 08 01 7F 80 18 01 7F 00 18 01 7B C0 3C 03 :62
63D0 77 CE 2C 03 7F FF 7F C3 7E FF F7 FF 7F FF FF FF :23
63E0 7D 9F FF C1 7F 8F F8 C1 67 0F E0 01 61 04 70 01 :D0
63F0 60 00 3C 01 60 00 7E 01 F0 00 FF 01 60 01 FF C1 :8D
-------------------------------------------------------
Sum: 41 18 7E 8A 66 A8 62 6E CA 5B B3 60 29 1D 5B 3A :52

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6400 40 00 3F 81 40 3F BF 01 40 3F BC 01 40 3F B8 01 :B3
6410 78 3F B0 39 70 5F DF F1 60 DF DF E7 7F DF DF FF :80
6420 7F DF DF F7 77 EF E3 C7 71 EF E1 87 70 6F E0 0F :DA
6430 70 2F E0 0F 78 17 F0 1F 70 17 F0 1F 40 17 F0 07 :10
6440 40 17 F0 03 70 0B F8 07 78 0B F8 1F 70 0B F8 1F :F0
6450 60 E5 FD 0F 47 F5 FD CF 7F FD FD FB 7E FE FE F9 :40
6460 6F FE FE F9 7F FF 7F 79 79 FF 00 31 58 7F FF C1 :1A
6470 78 3E 7F 81 68 1C 7F 01 7C 08 FE 01 7C 00 0F E1 :A9
6480 7E 00 1E C5 40 00 3F CF 60 00 FB DF 70 00 7E FF :D6
6490 7C 40 3F FF 7D 9F FF C1 7F AF F8 C1 67 6F E0 01 :74
64A0 69 74 70 01 6E F3 BC 01 6F DF 7E 01 76 BE FF 01 :6D
64B0 6F 7D FF C1 5E 80 3F 81 7D 80 3F 01 7F 80 3C 01 :C3
64C0 47 80 38 01 7B 80 30 39 77 40 1F F1 60 C0 1F E7 :51
64D0 7F C0 1F FF 7F C0 1F F7 77 E0 0B C7 71 E0 0D B7 :F0
64E0 70 60 0E 6F 70 20 0E EF 78 10 07 DF 70 10 05 5F :2C
64F0 40 10 06 E7 40 10 06 FB 70 08 02 87 78 08 03 DF :F1
-------------------------------------------------------
Sum: 76 66 4F 28 70 41 00 54 0E 79 42 9A B6 91 38 AE :E8

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6500 70 08 02 DF 60 E4 01 2F 47 F4 01 CF 7F FC 01 FB :4F
6510 7E FE 00 F9 6F FE 00 F9 7F FF 00 79 79 FF 00 31 :7B
6520 5A 7F FF C1 7B BE 7F 81 6B DD 7F 01 7D EA FE 01 :00
6530 7D B6 0F E1 7E DF DE C5 40 C3 3F CF 6F F6 FB DF :73
6540 73 BB 7E FF 7D 5F BF FF 7F FF FF FF 7F FF FF FB :39
6550 6F FF FF BF FD FF FF FF 7F FF FF EF 7F FF FF FD :0C
6560 7F FF 7F FF 77 F8 1F FF 7F FE 3F FF 6F FF FF F3 :A4
6570 7F FE 00 73 7F FE 00 61 7B FE 00 41 70 FE 00 41 :37
6580 70 7E 00 43 70 04 00 97 60 04 00 BF 60 04 00 FF :C2
6590 70 04 00 8F 78 0C 00 87 68 08 01 87 78 08 01 03 :8A
65A0 7C 18 01 01 56 38 01 01 7F F0 01 01 7F F0 03 87 :90
65B0 71 F0 03 FF 70 20 03 FD 60 20 07 F9 40 20 07 F9 :D3
65C0 40 40 07 F9 40 40 0F F9 40 80 1F F1 60 80 1F E1 :B8
65D0 7F 00 1F E1 7E 00 3F 81 7C 00 7F 81 7F FC 3C 61 :51
65E0 7F F8 10 3F 77 F8 00 07 6D FC 00 07 7F FC 00 07 :2E
65F0 7B DE 00 3F 6F 7F 80 FF 7F FF E7 FF 7F FF FF FF :E5
-------------------------------------------------------
Sum: 2B 92 46 D4 8A F2 0D 68 B8 24 8A FE 35 69 5C 02 :28

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6600 7F FF FF FB 6F DF 7F BF 7D EB 27 FF 7F F1 47 EF :38
6610 7F 84 5B FD 7F 82 07 FF 77 B8 1F FF 7F FE 3F FF :6A
6620 6F FF FF F3 7F FE 00 73 7F FE 00 61 7B FE 00 41 :E8
6630 70 FE 00 41 70 7E 00 43 70 04 00 97 60 04 00 BF :0E
6640 60 04 00 FF 70 04 00 8F 78 0C 00 87 68 08 01 87 :69
```

```
6650 78 08 01 03 7C 18 01 01 56 38 01 01 7F F0 01 01 :1B
6660 7F F0 02 87 71 F0 00 7F 70 20 00 8D 60 20 01 01 :77
6670 40 20 04 11 40 40 00 49 40 40 03 21 40 80 00 01 :A3
6680 60 80 03 01 7F 00 00 81 7E 00 12 01 7C 00 11 01 :03
6690 7F FC 00 61 7F F8 00 3F 77 F8 00 07 6D FC 00 07 :78
66A0 7F FC 00 07 7B DE 00 3F 6F 7F 80 FF 7F FF E7 FF :EB
66B0 FF C7 FF FF 73 8C 7B F7 6D 9B BF 7F 6D 37 DF FF :FD
66C0 73 07 C3 EF 6E 3F FD FF 7E 3E FD FB 7C 3C 7D DF :9D
66D0 78 49 23 FF 58 F6 DF FF 78 F6 DF F7 78 F9 3C FF :FF
66E0 78 00 3B 7F 64 3F B7 BF 58 3F A7 8F 5A 3F BF F7 :67
66F0 67 3D BE FB 7F D0 DC 7B 7F D8 D9 27 6F D5 56 DF :D3
-----------------------------------------------------------
Sum: 9B 68 41 96 0F CF 71 FA FF A6 F7 5A F2 04 2E 32 :6F

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6700 7E 59 56 DF 7D AE E9 3B 7B CF 69 FF 73 CF EC FF :3A
6710 6F F4 14 7F 6F B5 56 33 6F 16 57 01 72 4B BA 01 :F8
6720 7D B3 9B 03 7D B5 3D 07 7E 4D 1D 0F 7F F8 EE 9F :3F
6730 7C F6 EA BF 7B 76 63 7B 5B 39 7F BF 5C 1E 80 1F :D5
6740 4F 0C FF FF 4F 80 07 F7 47 C0 0F FF 60 E0 1F DF :79
6750 63 40 3F FB 73 00 3E DF 78 00 7B FF 7C 00 FF 77 :51
6760 7C 01 FF FF 7F C7 FF FF 73 8C 7B F7 61 98 3F 7F :E7
6770 61 30 1F FF 73 00 03 EF 6E 00 01 FF 7E 00 01 FB :FC
6780 7C 00 01 DF 78 49 23 FF 58 F0 1F FF 78 F0 1F F7 :23
6790 78 F9 3C FF 78 00 38 7F 64 00 30 3F 40 00 20 0F :1D
67A0 42 00 00 07 67 00 00 03 7F C0 00 03 7F C0 19 27 :74
67B0 6F C0 10 1F 7E 40 10 1F 7C 20 09 3B 78 00 09 FF :AB
67C0 70 00 0C FF 60 00 04 7F 60 00 06 33 60 00 07 01 :5F
67D0 72 48 02 01 7C 00 03 03 7C 04 01 07 7E 4C 01 0F :A1
67E0 7F F8 00 9F 7C F0 00 BF 78 70 00 7B 58 39 00 3F :74
67F0 5C 1E 80 1F 4F 0C FF FF 4F 80 07 F7 47 C0 0F FF :54
-----------------------------------------------------------
Sum: D7 8A 26 DA 14 5A 97 94 BD 7B C8 EA A7 9D EA 08 :1A

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6800 60 E0 1F DF 60 40 3F FB 70 00 3E DF 78 00 7B FF :97
6810 7C 00 FF 77 7C 01 FF FF 00 00 00 00 80 00 00 00 :ED
6820 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6830 08 00 00 00 00 00 08 00 00 00 08 00 03 F8 00 00 :0B
6840 03 F8 00 00 03 B8 00 00 02 08 00 00 01 0C 00 00 :CD
6850 00 B4 00 00 01 34 00 00 01 EC 00 00 00 C6 00 00 :9C
6860 00 AA 00 00 10 CB 00 00 00 73 00 00 00 77 80 00 :EF
6870 00 61 80 00 00 35 C0 00 00 38 C0 00 00 1C 60 00 :4A
6880 00 1F 60 00 00 0C 70 00 00 0D F0 00 00 06 F8 00 :F6
6890 00 07 6C 00 00 03 8C 00 00 03 FE 00 00 01 FF 00 :03
68A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
68B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
68C0 00 00 00 00 00 00 00 00 00 00 00 00 7F FF FF FF :7C
68D0 7B DF 7B EF 6F FD EF BB 7E EF FD FF 77 FF FF F7 :AF
68E0 7F FF FF DF 6F FF F7 FF 7F FF EB FB 78 03 EB FF :89
68F0 78 03 A2 FF 78 03 C1 FD 78 03 41 7F 78 03 80 FB :86
-----------------------------------------------------------
Sum: 51 9E 86 23 46 3B A9 B1 E8 A0 1D 58 E2 68 BB EF :64

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6900 7C 01 80 BF 7C 01 00 7F 7C 01 00 7F 7C 00 00 5F :8F
6910 7E 00 00 3F 6E 00 00 3F 56 00 00 3F 45 00 00 2F :73
6920 43 00 00 1F 43 00 00 1F 42 80 00 1F 41 80 00 17 :7D
6930 41 C0 00 0F 41 40 00 0F 40 E0 00 0F 40 E0 00 0F :FE
6940 40 F0 00 0F 40 F0 00 0F 40 78 00 07 40 78 00 07 :FC
6950 40 00 00 03 40 00 00 01 40 00 00 01 40 00 00 01 :06
6960 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
6970 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
6980 FF FF FF FF 7F FD FB FB 6F CF FE FF 7F B7 CF CF :7D
6990 5C 7B B7 8D 7D CC 78 9F 7D 85 CE BF 7E CD 86 FB :D6
69A0 7E 7A CD FF 7D B7 7B FF 63 C6 FE F9 6E 74 01 13 :88
69B0 6C 35 FD D3 76 6D FD DF 7B DD 3D BF 77 ED AD FF :94
69C0 7A 5A 0B 3F 7D BB 6A DF 7F FB D9 E3 7F FB FB 3B :85
69D0 77 F6 F6 1B 63 F4 F7 37 73 F6 35 EF 7A 6D AB F7 :19
69E0 7D AB 6D 2F 63 EB EE DF 6E 5B DF FF 6C 5A DF CF :FA
69F0 76 BA BF 8F 7B BD B3 9F 77 7F 6D BB 7B 7F 1E 3F :7D
```

```
-----------------------------------------------------------
Sum: A7 8F 2D B6 1B 75 ED 0A F5 9B 61 F8 04 FE A6 DA :0B

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6A00 7C 82 73 B7 7E AA 61 BF 7D DD B3 7B 7B FD DE FF :4D
6A10 70 03 BF 7B 7B DF D2 FF 77 EF ED 77 7A 5F DF 3F :99
6A20 75 BF 9F 1F 67 FF 1F 9B 47 DF BB FF 4F FB FE F7 :31
6A30 7F FF FF FF 7F FF FF FF 7F FD FB FB 6F CF FE FF :A5
6A40 7F 87 CF CF 5C 03 87 8D 7C 00 00 9F 7C 00 00 BF :6D
6A50 7E 00 00 FB 7E 02 01 FF 7C 07 03 FF 60 06 00 F9 :DD
6A60 60 04 00 13 60 04 00 13 70 0C 00 1F 78 1C 01 BF :DD
6A70 70 0C 01 FF 78 18 03 3F 7D B8 02 1F 7F F8 00 03 :1E
6A80 7F F8 00 03 77 F0 00 03 63 F0 00 07 73 F0 04 0F :B4
6A90 7A 60 08 07 7C 20 0C 0F 60 00 0E DF 60 00 1F FF :6B
6AA0 60 00 1F CF 70 00 3F 8F 78 00 33 9F 70 00 61 BB :62
6AB0 78 00 00 3F 7C 00 00 37 7E 00 00 3F 7C 01 80 7B :9F
6AC0 78 01 C0 FF 70 03 80 7B 78 1F C0 FF 70 0F ED 77 :DF
6AD0 78 1F DF 3F 75 BF 9F 1F 67 FF 1F 9B 47 DF BB FF :A7
6AE0 4F FB FE F7 7F FF FF FF 00 00 00 00 00 00 00 E0 :9B
6AF0 80 00 01 F0 00 00 3F F0 00 03 FF E0 00 01 FF F0 :72
-----------------------------------------------------------
Sum: 3D 4D 65 69 D4 79 84 97 37 84 7A 06 FC 20 65 38 :B4

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6B00 00 00 07 F8 00 0E 0F F8 01 FC 38 3C 1F F8 30 1C :E8
6B10 0F FE 40 0C 00 FF 80 04 03 9F C7 00 1E 00 0F 80 :F2
6B20 18 03 FF 00 20 07 FE 00 00 0F FE 00 00 00 7F 00 :CB
6B30 00 00 DF 00 00 07 87 80 00 06 01 80 00 08 00 00 :7C
6B40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6B50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6B60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6B70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6B80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6B90 00 00 00 00 00 00 00 00 00 00 00 00 7F FF FF FF :7C
6BA0 7F FF FF 1F 7F FF FE EF 7F FF C1 EF 7F FC 3F DF :CE
6BB0 7F FE 07 EF 7F FF FB F7 7F F1 F0 37 7E 0B C7 DB :A5
6BC0 61 F7 CF EB 70 39 BF F3 7F 1E 7F FB 7C 60 38 FF :97
6BD0 61 FF F7 7F 67 FC 0A FF 5F FB FD FF 7F F0 7D FF :83
6BE0 7F FF 9E FF 7F FF 26 FF 7F F8 79 7F 7F F9 FE 7F :22
6BF0 7F F7 FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :E8
-----------------------------------------------------------
Sum: E5 EA 8E 7A F3 4C FB 52 DE B0 A3 5A B2 4E 75 D1 :34

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6C00 7F FF FF FF 7F FF FF FF 7F C0 0F FF 7C 00 00 FF :C0
6C10 60 00 00 0F 40 00 00 03 40 00 00 01 40 00 00 01 :34
6C20 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
6C30 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
6C40 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
6C50 7F FF FF FF 7F FF FF FF 7F FF FF FF 7F FF FF FF :F0
6C60 7F FF FF FF 7F FF FF FF 7F FF EF FF 7F FF EF FF :D0
6C70 7F FF E7 FF 7F FF CF FF 7F FF EF FF 7F FF EF FF :88
6C80 7F FF EB FF 7F FF EF FF 7F FF FF FF 7F FF FF FF :CC
6C90 7F 7F FF FF 7F 7F FF FF 7E BF FF FF 7F FF EF FF :9F
6CA0 7F FF E0 5F 7F FF C0 3F 7F FF 80 1F 7D 3F 80 0F :A2
6CB0 70 1F 80 07 70 07 00 07 60 00 00 01 40 00 00 01 :36
6CC0 40 00 00 01 40 00 00 01 40 00 00 1F 40 00 00 1F :40
6CD0 40 00 00 1F 7C 20 00 3F 7F F0 00 3F 7F FC 00 3F :A2
6CE0 7F FF 80 7F 7F FF FC FF 7F FF FF FF 60 7D FB FF :49
6CF0 7F 7B 81 FF 7C 73 FB FF 7D EB F3 07 7D FB EB FF :27
-----------------------------------------------------------
Sum: 87 12 2F 10 A0 12 71 84 93 54 5C 82 50 AE 31 6A :DD

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6D00 7B FB DB FF 7F FF FF FF 7F FF FF FF 7F FF 3F FF :04
6D10 7F FF 1F FF 7F FF 9C 7F 7F FF D8 FF 7F FE 69 FF :6F
6D20 7F FE 2B FF 7F FF A6 7F 7F FF CC 7F 7F FF EC FF :7C
6D30 7B FE 6D FF 73 CE 2B FF 71 8F AF FF 79 9F C1 FF :D6
6D40 7D BE 03 FF 62 7F 07 E7 71 7F 83 E3 7E BF 01 E7 :87
6D50 7F 06 07 EF 78 0F E0 5F 7C 1F C0 3F 7E 0F 80 1E :07
6D60 7C 27 80 0F 70 1F 80 07 70 07 00 07 60 00 00 01 :27
```

ダンプ・リスト

```
6D70 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 1F :22
6D80 40 00 00 1F 40 00 00 1F 7C 20 00 3F 7F F0 00 3F :47
6D90 7F FC 00 3F 7F FF 80 7F 7F FF FC FF 7F FF FF FF :2C
6DA0 60 7D FB FF 7F 7B 81 FF 7C 73 FB FF 7D EB F3 07 :9C
6DB0 7D FB EB FF 7B FB DB FF 00 00 00 00 80 00 00 00 :32
6DC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6DD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6DE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6DF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
--------------------------------------------------------
Sum: 48 55 02 56 33 ED AF E6 02 C3 8C E3 8D 43 C8 67 :DD

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6E00 00 00 00 00 00 00 00 00 00 07 E0 00 00 0F C0 00 :B6
6E10 00 8F C0 00 00 CF 80 00 03 FF 00 00 03 FD 00 00 :A0
6E20 07 DE 00 00 07 1B 00 00 0C 0D 00 00 08 04 80 00 :AC
6E30 00 04 80 00 00 02 40 00 00 02 A0 00 00 00 A0 00 :08
6E40 00 00 10 00 00 00 00 00 00 00 08 00 00 00 00 00 :18
6E50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6E60 00 00 00 00 00 00 00 00 00 00 00 00 7D F6 DF BB :0D
6E70 62 ED AF 77 77 03 78 EB 73 01 F0 1D 6C 00 E0 1F :3E
6E80 5C 00 E0 0F 78 00 60 07 78 00 60 07 70 00 70 03 :EC
6E90 70 00 F0 03 60 00 F8 01 60 01 F8 01 60 01 FC 01 :74
6EA0 60 04 FA 01 60 1D 77 01 60 3C F7 01 60 79 67 01 :29
6EB0 60 F5 DB 01 70 FF F9 03 71 F8 0A 03 71 B7 ED 05 :2C
6EC0 63 6F DD 05 52 2F D9 09 74 0F B5 09 7B 3F 7D 09 :98
6ED0 7B E1 7D 11 77 C0 FF 11 77 00 7F 11 6C E0 7E 31 :33
6EE0 6B F0 3E 31 67 F0 3E 21 7F F8 1E 21 7F F8 8E 61 :9C
6EF0 7F FC 8E 61 5F FE 46 61 6F FE 67 61 7F FF 63 61 :E5
--------------------------------------------------------
Sum: BD 93 CA 33 B5 E8 5C 93 04 50 8A C5 7A 4D 4B E0 :6E

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6F00 5F FF 33 61 6F FF B9 E1 7F FF B9 E1 5F FF FD E1 :4E
6F10 6F FF FD E1 7F FF FF E1 7F FF FF E1 7F FF FF E1 :66
6F20 00 00 00 00 00 00 00 00 80 00 00 00 00 80 00 00 :00
6F30 00 80 30 00 01 80 60 00 01 C0 C0 00 03 C1 C0 00 :96
6F40 03 C3 C0 00 01 E7 E1 00 01 FB E2 00 00 FB F7 00 :1F
6F50 00 7D F7 80 00 71 F4 00 00 0D EE 00 00 7F FD 00 :D0
6F60 01 BF FB 80 01 FF E0 00 00 FF 80 00 00 30 00 00 :CA
6F70 00 3E 00 00 00 3C 00 00 00 3C 00 00 00 FE 00 00 :B4
6F80 00 7F 00 00 00 00 00 00 00 00 00 00 00 00 0C 00 :8B
6F90 00 00 1E 00 00 00 0D 00 00 00 00 2C 00 00 30 1E :A5
6FA0 00 00 78 0C 00 00 34 00 00 00 00 00 00 00 60 C0 :D8
6FB0 00 00 F1 E0 00 00 68 C0 00 00 00 00 00 00 02 00 :FB
6FC0 00 00 07 00 00 00 07 00 00 00 03 00 00 00 01 00 :12
6FD0 00 00 00 00 7F FF FF FD 7F BF FF BB 7F 3F FF 11 :40
6FE0 7E 3F 83 2D 7E 3F 37 EF 7C 3E 6F C7 7C 1C CF C7 :6E
6FF0 78 19 DF 6F 78 13 DE 7F 7C 07 ED 7F 7C 03 EA F9 :18
--------------------------------------------------------
Sum: C8 92 02 CA 66 62 91 ED F7 05 26 EF 58 45 07 71 :92

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7000 7E 03 F7 71 7F 01 F7 B9 7F 81 F4 3F 7F 8D EE FF :45
7010 7E 7F FC 1F 7D BF F8 3F 79 FF E0 7D 66 FF 9F FB :5F
7020 7F 30 7F 9B 7F BE FF 07 7F BD FF 97 7F 3D FF F5 :8E
7030 7E FE FF E1 7E 7F 3E 0D 7F 00 1F 2F 7F FF FF E9 :D7
7040 7F FF FF D1 6E FF FF DB 64 7F 1F BF 70 78 FF 7F :BC
7050 70 47 FF 7F 60 BF FE FF 70 7F FE FF 7C 7F FD FF :34
7060 7A FF FD FF 47 FF FB FF 77 FF F3 FF 6E FF FB FF :84
7070 6F 7F FF FF 77 7F FF FF 76 FF FF FF 79 FF FF FF :C9
7080 7F FF FF FF 7F FF FF FF 76 6F BD 77 75 6F 8B 3B :BB
7090 76 9F A8 D7 78 9F D7 7B FF 7F C0 FF 7E FF FF 87 :3D
70A0 5F FF FE 3F 4F FF FE 7F 47 FF FC 7F 43 FF F8 FF :60
70B0 67 FF C4 FF 6C 7F BA FF 79 3F 7D FF 7B 9A FD FF :12
70C0 77 C4 FD FF 77 FF FD FF 77 FF FD FF 50 FF A3 FF :0C
70D0 4E 1F 3F FF 67 CE 43 FF 66 70 83 FF 73 1D 87 FF :90
70E0 71 03 86 1F 71 83 86 1F 70 81 06 0F 40 81 06 0F :8E
70F0 70 01 E4 0F 4C 02 34 07 63 02 DC 27 78 C2 10 23 :C2
--------------------------------------------------------
Sum: 32 F7 7A 9A D2 A7 AB 00 9C 57 59 66 E2 23 40 44 :9C

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7100 5F 32 C4 33 40 8C 1F 33 47 64 7C 0B 40 71 F1 E1 :5B
7110 47 47 E7 81 47 1F 82 01 44 7E 3C 01 41 F8 80 01 :98
7120 47 E3 BC 01 4F 8E 3C 01 7E 38 3C 01 78 E0 3C 01 :89
7130 63 80 3C 01 4E 00 3C 01 7C 00 3C 01 70 0F 80 01 :64
7140 70 0F 80 01 70 1F 80 01 78 1F C0 79 7F 7F C0 FF :9D
7150 7E FE FF 87 5F FC 7E 3F 4F BE 7E 7F 47 36 7C 7F :9C
7160 43 33 78 FF 67 B3 C4 FF 6C 73 82 FF 78 3F 01 FF :E1
7170 78 1A 01 FF 70 04 01 FF 70 00 01 FF 70 00 01 FF :E6
7180 50 00 23 FF 4E 00 3F FF 67 C0 43 FF 66 70 83 FF :BF
7190 73 1D 87 FF 71 03 86 1F 71 83 86 1F 70 81 06 0F :CE
71A0 40 81 06 0F 70 01 E4 0F 4C 02 34 07 63 02 1C 27 :6B
71B0 78 C2 10 23 5F 32 04 33 40 8C 1F 33 70 64 7C 0B :AE
71C0 70 71 F1 E1 70 47 E7 8F 70 1F 82 3D 70 7E 00 FD :19
71D0 71 F8 81 F9 67 E3 81 F9 4F 8E 01 F1 7E 38 81 E1 :8E
71E0 78 E3 81 E1 63 80 3C 01 4E 00 3C 01 7C 00 3C 01 :21
71F0 3F FF FF 80 3F DF 9E 00 B0 91 C2 00 20 00 00 00 :9C
--------------------------------------------------------
Sum: 0C E1 4D A7 D1 CA CB 5D 49 79 8E 8B 4A 59 49 7F :EA

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7200 30 00 00 04 3C 21 00 0E 30 73 80 0E 00 73 80 04 :C7
7210 00 31 00 30 00 08 00 78 00 00 00 30 00 00 00 00 :11
7220 00 20 00 00 00 70 00 00 00 70 00 00 00 20 00 00 :20
7230 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7240 04 00 00 00 04 00 00 00 01 C0 00 00 01 BC 00 00 :86
7250 01 8E 80 00 00 CF 70 00 00 7E F4 00 00 7F 6E 00 :AD
7260 00 7E F7 00 00 3F EF 00 00 0F FE 00 00 00 FF 00 :AF
7270 00 00 7E 00 00 84 3F 00 01 CE 5E 80 01 CE E6 00 :A3
7280 18 84 E4 00 3C 00 40 00 18 00 00 00 01 10 00 00 :25
7290 03 B9 00 00 03 BB 80 00 01 13 80 00 00 01 00 00 :8F
72A0 00 00 00 00 40 00 00 7F 40 00 00 7F 40 00 00 FF :BD
72B0 4E 6E 29 FF 41 7F FF FB 41 DE FF F1 43 8C 77 F1 :E4
72C0 4F 8C 73 FB 7F CE E3 CB 7F F6 E3 83 7F F0 F7 CB :50
72D0 7F C6 1F FB 7F DF C7 FB 7F 8F B0 7B 7F 8E 7F 0B :4F
72E0 7F DC 63 E3 7F FC 67 F9 7F FC 47 FF 7F FC CC 7F :03
72F0 7B FF FE 39 75 FF FE 31 74 1F FF B1 79 C1 FF CF :9F
--------------------------------------------------------
Sum: 66 35 F5 45 F2 0D 6C F0 BD 8F 28 DC 7C 74 8B 18 :13

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7300 7D BC 3F 07 7D 8E 86 33 7C CF 71 79 7C 7E F4 C5 :2B
7310 71 7F 6E 43 77 7E F7 27 72 3F EF 1F 7C 8F FE CF :4B
7320 7D E0 FF 67 79 FE 7E F3 7B 7B 3F 7F 7A 31 9E BF :67
7330 7A 31 06 7F 67 7B 14 73 40 07 BC 63 66 F0 7C E3 :B4
7340 7E EF 01 63 7C 46 F8 2F 7C 44 63 1F 7E EC 47 C7 :74
7350 7F FE EF F1 7F FF FF F9 00 00 00 00 80 00 00 00 :53
7360 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7370 00 00 00 00 00 00 00 00 00 00 00 00 00 00 38 00 :38
7380 00 00 7E C0 00 03 BD E0 00 03 80 F0 00 07 3A 78 :0A
7390 00 00 43 34 00 06 F7 86 00 0D 29 CE 00 0D 55 B6 :16
73A0 00 0C 84 36 00 02 3D B6 00 05 C1 D2 00 02 F7 40 :8C
73B0 00 00 6E 00 00 00 00 00 00 00 00 00 00 00 00 00 :6E
73C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
73D0 04 00 00 04 07 40 00 1C 03 5C 00 74 07 50 01 78 :0E
73E0 06 B8 06 BC 00 30 0F 5C 00 00 3B 80 20 00 0D A0 :A3
73F0 20 00 03 80 36 00 01 00 0D 00 00 00 21 60 00 00 :68
--------------------------------------------------------
Sum: 0C FD 5E EE 0C 45 07 7C 35 45 63 1D 1E E0 1F 83 :C3

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7400 3F 7E 10 00 3C 40 FF 00 3F DF CE DC 7F FF FF FF :8C
7410 7F FF FF F3 7F FF FE C3 7F DF FC 83 7F CD F8 01 :D1
7420 7F C0 F8 01 7F 80 70 01 7F 80 70 01 7F 80 00 01 :18
7430 7E 00 00 03 7F 00 00 03 7F 80 00 07 7F E0 00 03 :6B
7440 7F F0 00 03 7F F0 00 01 7F E0 00 01 7F E0 00 01 :A2
7450 7F 00 00 01 7E 00 00 01 7C 00 00 01 7E 00 00 01 :FB
7460 7F 00 00 01 7F 00 00 01 72 00 00 01 78 00 60 01 :4C
7470 7C 00 C1 81 7E E3 C1 C1 7F F7 81 C7 7F FF F3 EF :BF
7480 73 FF FF F3 74 BF FF E5 77 00 FD 9D 73 1D F8 75 :89
7490 77 13 F8 79 70 3B E6 3D 70 07 CF 01 5F C7 BB 83 :74
```

```
74A0 4F FF 8D 0F 49 FF F3 1F 40 7F FC 7F 40 1F FE FF :DA
74B0 40 01 CF FF 40 00 00 FF 40 00 00 03 40 00 00 01 :D2
74C0 00 00 00 0E 80 00 00 06 00 00 00 06 00 08 00 02 :A4
74D0 00 18 00 02 00 3C 00 00 01 FF 00 10 00 7F 00 12 :F7
74E0 00 34 00 6A 00 20 00 7A 00 00 01 FA 04 00 00 7C :B3
74F0 04 00 03 B0 02 00 07 A0 07 00 03 C0 04 C0 0F C0 :BD
----------------------------------------------------------
Sum: 31 8B 1E 21 A2 E7 0D EB 17 1A 87 21 4A 55 0A 3E :3C


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7500 07 F0 1F C0 07 80 07 00 07 80 07 00 0F 00 02 00 :03
7510 0E 00 00 00 0E 00 00 00 0E 00 00 00 07 00 00 00 :31
7520 07 AA A0 00 07 AA AF 00 07 FF FF 02 03 FF FE 02 :BA
7530 03 FE FC 06 01 6F 7C 06 01 5E 7C 0E 01 41 78 0E :A6
7540 01 40 B8 1E 01 40 58 1E 02 80 28 3E 01 00 14 36 :01
7550 01 00 30 68 01 00 20 1E 01 00 20 7E 01 00 20 7C :14
7560 00 00 00 BA 00 00 03 0A 00 00 07 F2 00 00 0F FA :C9
7570 00 00 1F FE 7F FF FF 8F 7F FF FF 87 7F F7 FF 87 :29
7580 7F E3 FF 83 7F C3 FF 83 7E 00 FF 81 7C 00 E7 01 :0A
7590 7E 00 C7 01 7F 80 01 01 7F 8A 00 01 6B 9F 00 01 :5C
75A0 75 FE 00 01 75 FE 00 03 72 FC 20 0F 77 3F E0 1F :3C
75B0 74 C7 E0 1F 77 F7 C0 1B 77 87 80 13 77 BF F0 51 :8B
75C0 6F 7F F8 81 6E FF FD 81 6E FF FF C1 6E 7F FF 81 :EC
75D0 77 00 0F 01 77 AA A0 61 77 AA AF 41 77 FF FF 4B :7A
75E0 7B FF FE FB 7B FE FD F7 7D 6E 7D F7 7D 5E 7D EF :86
75F0 7D 41 7B EF 7D 5E BB DF 7D 5F 5B DF 7A BF AB BF :56
----------------------------------------------------------
Sum: E5 3F E8 14 65 15 C1 35 64 DF F5 C1 4C 6F 97 2F :0A


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7600 7D 3F D5 B7 7D 3F B3 61 7D 3F A3 01 7D 3F A6 01 :DB
7610 7D 3F A6 01 78 00 00 83 40 00 03 03 40 00 07 F3 :DE
7620 40 00 0F FB 40 00 1F FF 00 3B FB FE 80 31 81 3A :48
7630 00 78 00 10 00 00 00 32 00 00 70 02 00 00 FE 02 :2C
7640 00 1F FF C6 00 0F DF 86 00 07 8F 0C 00 00 00 00 :FA
7650 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7660 00 00 00 00 00 00 00 00 00 00 00 00 07 C0 00 00 :C7
7670 0F F8 00 00 1F F8 00 00 3F F8 00 00 3F F8 00 00 :8C
7680 3F FC 00 00 3F FE 00 00 3F 00 00 00 3F 7C 80 00 :F2
7690 3F 7E 40 00 3F 3F 30 00 3F 3E 1A 00 3E 1C CB 00 :67
76A0 3C 19 0D 80 00 0F D6 80 00 13 DA C0 00 0C FC C0 :BC
76B0 00 03 1F 60 00 0C E1 E0 00 03 00 00 00 00 C0 00 :12
76C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
76D0 00 00 00 00 00 00 00 00 00 00 00 00 7F 80 00 01 :00
76E0 7F 80 04 01 7F 02 7C C1 7F 87 FF 81 7F FF BF C1 :46
76F0 7F FF 37 F9 7F FF 66 F1 7F F1 5C F1 7F F8 8B E1 :23
----------------------------------------------------------
Sum: 01 22 30 63 D0 9F 7A AD 78 45 EF 42 7D 43 7D 93 :0A


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7700 73 FF 07 F3 40 FF 8F FF 40 7F 1F FF 40 0F 1F FF :83
7710 40 07 3F FF 40 02 0F FF 40 00 07 FF 40 00 03 FF :5D
7720 40 00 03 FF 4D F8 00 FF 48 F8 01 FF 6C E0 03 FF :14
7730 76 30 07 FF 7F F8 83 FF 67 FD E1 FF 6F 00 71 FF :C8
7740 59 00 1D FF 79 00 07 FF 6D 80 01 FF 4C 80 00 FF :AC
7750 5F 00 00 7F 5F 80 00 3F 40 C0 00 3F 40 40 00 1F :DA
7760 40 60 00 1F 40 20 00 0F 40 40 00 0F 50 F0 00 0F :0C
7770 70 FC 1F FF 61 FF 1F FF 4F FF FF FF 58 7F FF FF :29
7780 40 00 FF FF 40 00 0F FF 40 00 00 FF 40 00 00 7F :8A
7790 3F E0 00 00 BF F0 00 00 3F 70 00 00 30 30 00 00 :DD
77A0 20 60 00 00 20 00 06 00 20 00 06 00 30 00 00 00 :FC
77B0 3F 80 00 00 18 00 00 38 28 00 00 7C 28 0F 01 F6 :E1
77C0 1C 37 81 E2 3C 1F 80 48 38 1F 80 36 00 0F E0 36 :0B
77D0 00 0E 30 00 00 0D F8 00 00 05 FC 00 00 05 FE 00 :47
77E0 00 05 FF 00 00 02 FF 80 00 03 7D C0 00 01 9E F8 :5C
77F0 00 01 EF 7E 00 00 F3 AE 00 00 3C 06 00 00 10 02 :63
----------------------------------------------------------
Sum: CB 9D 2A EB 38 AE C6 F5 6A 8A 43 BF 57 72 22 CD :CC


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7800 00 00 60 00 00 00 00 30 00 00 00 78 00 00 03 FE :09
7810 03 00 03 26 03 00 00 DA 00 00 00 5A 00 00 00 06 :69
7820 00 70 00 02 00 F8 00 00 07 DF 00 00 07 8F 00 00 :E6
7830 01 24 00 00 00 D8 00 00 00 D8 00 00 00 00 00 00 :D5
7840 00 00 00 00 40 0F FF FF 40 07 FF FF 40 07 8F FF :67
7850 40 07 C7 FF 4B 0F E1 FF 4B 1F F0 FF 4F FF F0 FF :DD
7860 40 7F F8 FF 40 3F FC 47 40 7F FE 03 43 E0 FE 01 :5A
7870 43 0F 7C 01 41 37 BC 01 41 DF BE 01 43 DF 9F 81 :25
7880 47 EF EF 81 7F EE 37 C1 7F ED FB E1 6F F5 FD C3 :77
7890 67 F5 FE 87 67 F5 FF 6F 67 FA FF BF 67 FB 7D C7 :70
78A0 63 FD 9E F9 73 FD EF 7F 73 FE F3 AF 73 FF 3C 17 :AD
78B0 59 FF 13 FB 49 FC 67 CD 61 F0 06 01 71 E0 04 01 :8D
78C0 78 E0 00 01 78 60 00 01 78 70 00 01 7C F8 00 01 :90
78D0 7C 0E 00 01 7C 07 80 01 78 00 E0 01 70 00 78 01 :D1
78E0 70 00 3E 01 78 00 07 81 7E 03 80 01 7E 03 F8 01 :2B
78F0 7F 27 FF 81 7F FF FF C1 00 00 0F FE 80 00 0F FE :FE
----------------------------------------------------------
Sum: 14 1E 79 A7 9C A6 AA 10 3B 83 0D 25 C0 1E 58 27 :9B
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7900 00 00 07 FE 00 00 07 FE 00 00 03 FE 00 00 03 FE :0C
7910 00 00 01 FE 00 00 00 7E 00 00 00 0E 03 00 00 00 :8E
7920 07 00 00 00 06 00 00 00 00 00 00 04 00 00 00 04 :15
7930 00 00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 :04
7940 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7950 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7960 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7970 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7980 02 00 00 00 00 00 00 00 00 00 10 00 00 00 10 00 :22
7990 00 00 10 00 00 00 00 00 00 00 00 00 00 00 00 00 :10
79A0 00 00 00 00 00 00 00 00 00 00 00 7F FF E0 01 :5F
79B0 7F FF E0 01 7F FF F0 01 7F FF F0 01 7F FF F8 01 :B4
79C0 7F FD F8 01 7F F3 FC 01 7F C7 FE 01 7C 2F FF 81 :54
79D0 78 5F FF F1 70 3F FF FF 70 FF FF FB 71 FF FF F5 :41
79E0 76 FF FF F5 7A 7F FF F5 7A 7F FF F1 7A 7F FF D1 :08
79F0 7A 7F FF E1 7A 7F FF E1 78 7F FF A1 70 E3 FF C1 :5C
----------------------------------------------------------
Sum: 6F D9 ED C9 68 2F F0 53 60 C3 FE 9F D8 8E E7 0C :F1


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7A00 73 1C FF C1 77 BF 3F C1 7B BF CF 41 7B BF F7 81 :81
7A10 7D DF FB 81 7E 6F E3 81 5F 97 E1 81 5F E3 81 81 :C5
7A20 4F F0 81 01 4F F2 01 01 4F F5 83 01 47 F5 D7 01 :E0
7A30 40 F6 DE 01 42 F6 FE 01 42 F6 C6 01 40 EE D6 01 :50
7A40 43 EE D0 01 43 EE D0 01 47 DE C0 01 46 1E F0 01 :3F
7A50 47 B8 F8 01 4F 7D F8 01 4F FB FC 01 4F FF FC 01 :4F
7A60 00 00 00 00 00 00 00 00 80 00 00 00 00 00 00 00 :80
7A70 00 60 00 60 00 F0 00 F0 03 9C 03 9C 03 0C 03 0C :FC
7A80 01 98 07 98 00 F0 1B F0 01 F8 3C 78 00 90 E7 50 :A7
7A90 00 00 C3 00 20 00 66 00 30 00 3C 00 1C 00 7E 00 :4F
7AA0 0C 0C 24 00 18 1E 00 00 30 73 80 C0 38 61 81 E0 :4F
7AB0 10 33 07 38 00 1E 06 18 00 3F 03 30 00 12 01 E0 :23
7AC0 00 00 03 F0 00 00 01 20 20 00 00 00 30 00 00 00 :64
7AD0 31 80 00 00 3F C0 00 00 3F FF C0 00 3F FF C7 80 :33
7AE0 3F FF C7 8E 3F FF FF 8E 3F FF FF FE 3F FF FF FE :D4
7AF0 3F FF FF FE 3F FF FF FE 3F FF FF FE 3F FF FF FE :EC
----------------------------------------------------------
Sum: D5 3C DF F2 0D 5B 6F EA C2 5D 71 C6 3A AE C0 9E :3F


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7B00 3F FF FF FE 1F FF FF FE 03 FF FF FE 00 7F FF FE :D1
7B10 00 07 FF FE 67 FF F3 FF 63 FF F1 FF 63 FF F9 FF :08
7B20 73 9F FD 9F 7B 0F 3F 0F 7B 01 1B 01 7B 01 9B 01 :2A
7B30 7B 01 D8 01 64 03 E4 03 4E 07 C0 07 4C 02 00 03 :0D
7B40 7E 06 00 47 5F 6E 00 6F 4F FF 00 FF 41 FF 81 F3 :08
7B50 41 F3 00 E3 41 E1 81 E7 43 00 1B 2F 47 00 3E 1F :D2
7B60 43 00 30 03 46 80 70 03 6C C0 F0 03 7C 80 78 07 :49
7B70 5C C0 FC 0F 43 ED E0 07 60 7F C4 0F 74 07 CE DF :18
7B80 75 80 3F FF 71 D0 00 7F 7F D7 C0 01 7F C7 D7 81 :A8
7B90 7F FF D7 AF 7F FF C7 AF 7F FF FF 8F 5F FF FF FF :60
7BA0 43 FF FF FF 40 7F FF FF 40 07 FF FF 40 00 3F FF :C0
7BB0 40 00 00 7F 40 00 00 01 40 00 00 01 40 00 00 01 :82
7BC0 40 00 00 01 40 00 00 01 FF FF FF FF 7F FF FF FF :FA
```

グラフィック花札こいこい

ダンプ・リスト

```
7BD0 7F C0 7F FF 7F 3F 9F FF 7E FF EF FF 7D FF F7 FF :F6
7BE0 7B FF FB FF 77 FF FD FF 77 FF FD FF 6F FF FE FF :C3
7BF0 6F FF FE FF 6F FF FE FF 6F FF FE FF 6F FF FE FF :AC
-------------------------------------------------------
Sum: A8 9B 8C 02 A3 57 46 9B 6B 1D 3E D1 D7 C9 9C 75 :F4
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7C00 6F FF FE FF 6F FF FE FF 77 FF FD FF 77 FF FD FF :BA
7C10 7B FF FB FF 7D FF F7 FF 7E FF EF FF 7F 3F 9F FF :AD
7C20 7F C0 7F FF 7F FF FF FF 7F FF FF FF 7F C0 0F FF :02
7C30 7C 00 00 FF 60 00 00 0F 40 00 00 03 40 00 00 01 :6E
7C40 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
7C50 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
7C60 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
7C70 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
7C80 40 00 00 01 40 00 00 01 40 3F 80 01 40 FF E0 01 :A2
7C90 41 FF F0 01 43 FF F8 01 47 FF FC 01 47 FF FC 01 :F2
7CA0 4F FF FE 01 4F FF FE 01 4F FF FE 01 4F FF FE 01 :34
7CB0 4F FF FE 01 4F FF FE 01 4F FF FE 01 47 FF FC 01 :2A
7CC0 47 FF FC 01 43 FF F8 01 41 FF F0 01 40 FF E0 01 :CF
7CD0 40 3F 80 01 40 00 00 01 40 00 00 01 40 00 00 01 :C3
7CE0 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
7CF0 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
-------------------------------------------------------
Sum: 0B F9 E0 08 EF F9 E0 18 DA 38 53 0C D2 F9 61 0A :73
```

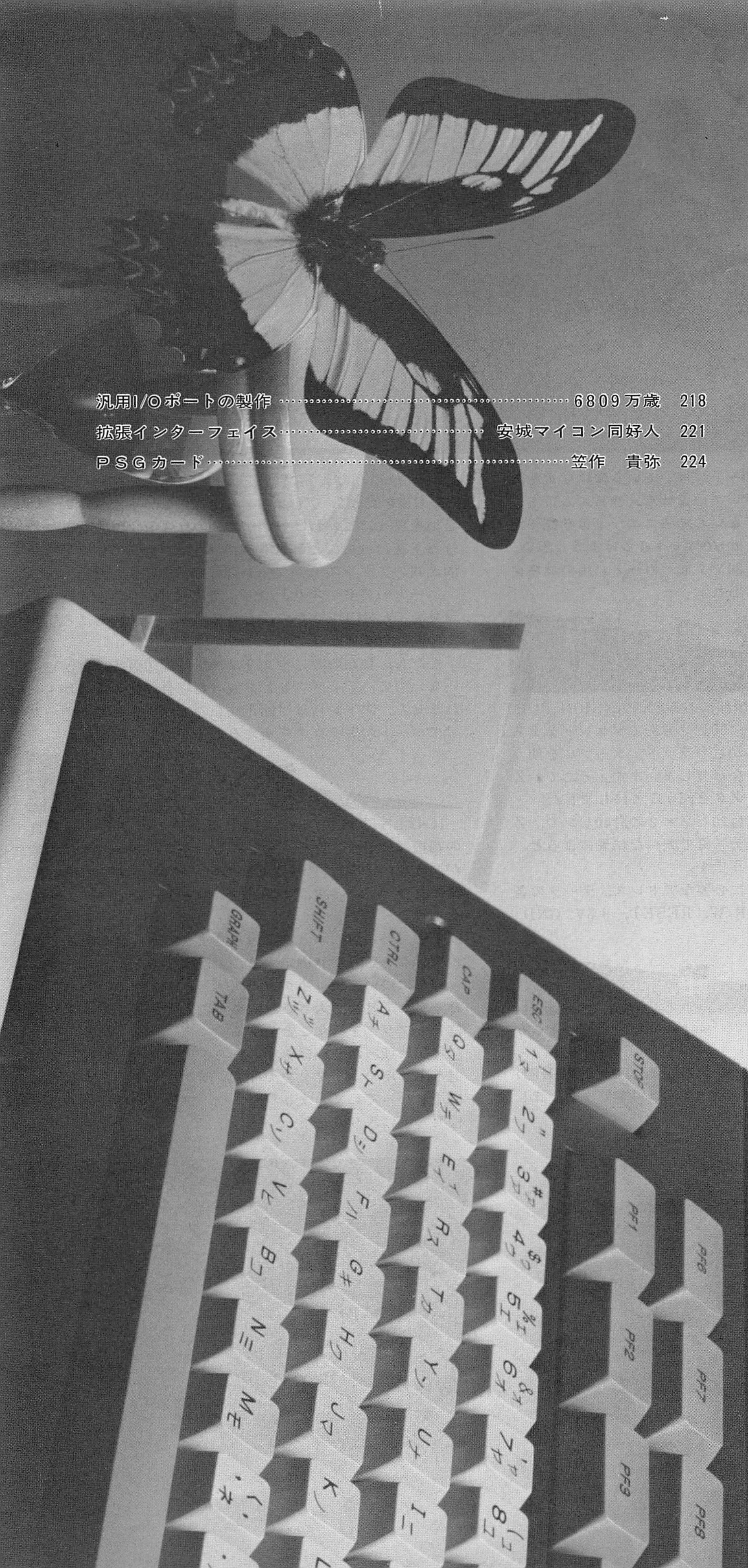
```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7D00 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
7D10 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
7D20 40 00 00 01 40 00 00 01 40 00 00 01 40 00 00 01 :04
7D30 7E 7F F3 FF 70 FF F8 FF 47 E3 FE 1F 7F EB FF C3 :C8
7D40 7F 0B FF F9 FF 79 FF FF 78 6C 7F FF 7B BF 1F FF :B2
7D50 7B F1 DF FF 63 79 C3 FF 69 FF E3 FF 6A FF EB FF :85
7D60 6A 0E 09 FF 6A 80 FC FF 6A AA FE 7F 4A A0 FF 3F :1E
7D70 7C 1F FE BF 5F FF FD 87 6F FF FB 71 6F FF F4 ED :63
7D80 77 FF E3 DD 77 FB FF BB 5B F1 FE 77 65 FB F9 CF :4B
7D90 78 F9 E4 3F 7F FB C3 FF 7F FF FF 3F 7F FF FE 07 :0F
7DA0 7F FF FC 01 7F FF FC 01 7F FB FC 01 7F FC F0 01 :D9
7DB0 60 30 00 01 40 00 00 01 40 00 00 01 40 00 00 03 :56
7DC0 40 00 00 03 40 00 00 03 40 00 00 07 70 00 00 07 :44
7DD0 7C 00 70 0F 7F 03 FC 1F 7F 1F FF 3F 7F FF FF FF :F0
7DE0 7F FF FF FF 7E 00 03 FF 70 00 00 FF 40 00 00 1F :CA
7DF0 40 00 00 03 40 00 00 01 40 00 00 01 40 00 00 01 :06
-------------------------------------------------------
Sum: 67 CE 0A EA 8D 68 70 64 C9 01 51 0E EF 3D E2 F0 :19
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7E00 40 00 00 01 40 00 00 01 40 00 00 01 48 00 00 01 :0C
7E10 48 00 08 01 48 00 08 01 48 00 FC 01 48 00 FE 01 :2E
7E20 48 00 FF 01 7C 1F FE 01 5F FC FC 01 4F 98 F8 71 :8A
7E30 4F 88 B0 E1 47 C9 23 C1 47 EA 7F 83 43 F1 FE 07 :C8
7E40 61 FA 38 0F 78 D8 60 3F 7F 8B C3 FF 5F 01 FF 3F :FB
7E50 49 83 FE 07 41 C7 9C 01 43 83 8C 01 41 1B EC 01 :12
7E60 41 FC F0 01 60 30 00 01 40 00 00 01 40 00 00 01 :41
7E70 40 00 00 03 40 00 00 03 40 00 00 03 40 00 00 07 :10
7E80 70 00 00 07 7C 00 70 0F 7F 03 FC 1F 7F 1F FF 3F :EB
7E90 7F FF FF FF 7F FF FF FF 03 80 00 00 07 00 00 00 :82
7EA0 8E 00 00 00 1C 00 00 00 38 00 00 00 30 00 00 00 :12
7EB0 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :20
7EC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7ED0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7EE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7EF0 00 00 00 00 00 01 E0 00 00 03 F0 00 00 03 7E 00 :55
-------------------------------------------------------
Sum: E7 00 DC 04 BB B7 74 16 2A 7A B2 A9 F8 C7 5C 01 :DE
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7F00 00 03 7E 00 00 03 5E 00 00 03 6E 00 00 02 F2 00 :47
7F10 00 01 FC 00 00 01 F0 00 06 01 F0 00 03 01 F0 00 :D9
7F20 07 00 00 00 07 00 00 00 07 00 00 00 07 00 00 60 :7C
7F30 03 00 00 00 00 00 04 60 00 00 03 60 00 00 03 60 :2D
7F40 00 00 01 A0 00 00 00 00 00 00 00 00 79 20 00 6F :A9
7F50 70 60 00 17 6A F0 00 0B 55 F8 00 07 68 0C 00 03 :17
7F60 50 06 00 03 60 02 00 01 40 03 20 01 70 03 30 01 :C4
7F70 78 01 B0 01 70 01 F8 41 70 01 FD 41 70 01 FF C1 :B4
7F80 60 09 E7 C1 60 0D 81 E1 40 0E 00 61 40 1C 00 21 :0C
7F90 40 10 77 E1 40 1F FB E1 41 1E 7B E1 41 3E 3D F1 :4B
7FA0 41 3E BD F1 43 BE 1E F1 63 BC 0E 71 63 F8 01 71 :A8
7FB0 73 F8 00 F1 71 F8 00 F9 79 F8 00 F9 7D F0 00 F9 :8E
7FC0 7F 68 00 F9 5F B4 00 F9 59 B8 07 FB 52 DC 07 FF :33
7FD0 51 3D F7 FF 62 7C 07 FF 75 7E DF FF 72 7F FF 9F :C8
7FE0 75 7F FF 2F 7A 7F F9 0F 68 7F F4 41 77 7F F1 21 :47
7FF0 7E FF 82 41 7F E0 01 21 7C 00 00 01 40 00 00 01 :7F
-------------------------------------------------------
Sum: 59 DD BE A7 4F 68 E5 81 21 95 E1 91 A7 4F 49 30 :4F
```

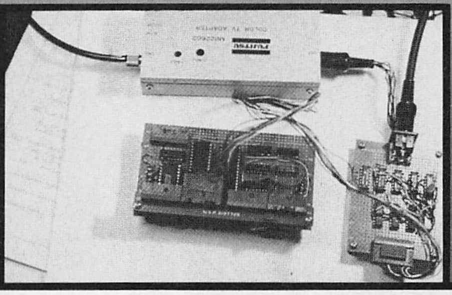
FM-8活用研究

ハード編



FM-8

汎用I/Oポートの製作



●6809万歳

FM-8には，アナログ・ポートが付いているのになぜか汎用のパラレル・ポートは付いていません．作るにしても，本体内にスロットは付いてないし拡張ユニットは発売予定のままです．しかし，何とかI/Oポートをつけようと思い，FM-8の基板上にある『Z80ソフト・カード』用のコネクタを利用することを考えました．

Z80ソフト・カード用のコネクタ

当初，I/Oポートは本体後部にあるEXPANSION PORTに接続するつもりでしたが，50ピンもあるソケットをテスタで調べるのは面倒で，しかもフラット・ケーブルを使うことになり，製作費が高くなってしまいます．そこで，Z80ソフト・カード用のコネクタを使うことにしました．

Z80カード用のコネクタは20ピン×2の計40ピンで，Z80カードの回路を見ながらテスタで調べた結果によると，**表1**のようになっているようです．

その40本の中にはPIAに必要なアドレス，データの各バス，計24本，さらにE，R/W，$\overline{\text{RESET}}$，+5V，GNDとすべて揃っています．

回　路

回路はI/O誌'81年4月号の『macaroni/Iの製作』を参考にしました．

問題になるのはポートのアドレスです．FM-8は&HFD00～&HFDFFまでの256バイトがI/Oエリアですが，詳しいマップは発表されていません．

macaroni/IのようにアドレスデコードにDIP・SWとEX-ORを使って，アドレスを可変式にしてトラブルのない番地を探せばよいわけですが，初めからPIAを2個つなげる予定だったことと費用の点から**LS138**を2個使って半固定式のアドレス・デコードにしてしまいました．

ポートのアドレスも上のほうは使っていないだろうという独断で&HFDE0からにしましたが，いまのところ実用上の問題はないようです．

ただし，現在のところプリンタ以外周辺装置は継がっていないので，ディスクなどを使った場合問題になるかもしれません．アドレスは半固定式なので配線を2本変えるだけでポートのアドレスを変えることができます．

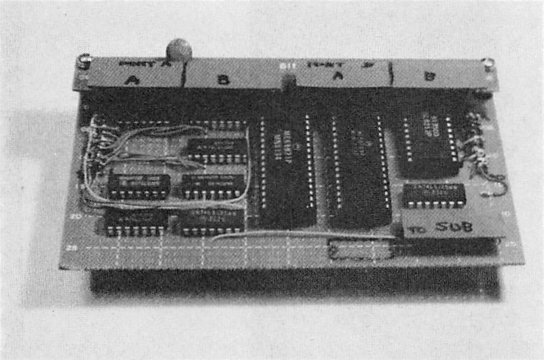
製　作

ICはLS-TTLが4個にLSI（PIA）が2個です（写真の基板は他のインターフェイスも付いているのでICの数が増えています）．PIAはFM-8のメインのCPUが**68A09**タイプなので，多少高くても**68A21**を使ったほうが良いでしょう．

工作で特に難しいところはありませんが，2つの20ピンのプラグをしっかり固定することが重要です．

部品についてはコネクタの間隔とFM-8の中のスペースを考慮して基板の大きさを決めることと，Z80用コネクタに合うプラグを探すことです．

実際に使った基板は普通の穴あ　　クでしたが，配線引き出し用のコネクタをつける場所がなくて，2階建ての構造になってしまいました．また，プラグのほうは数百円程度で手に入りました．



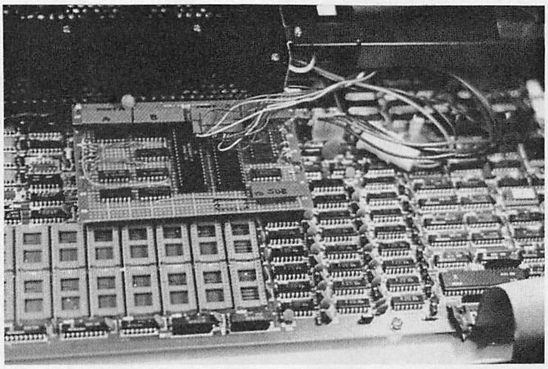


図1　I/Oポートの回路図（アドレスはFDE 0～FDE 7）

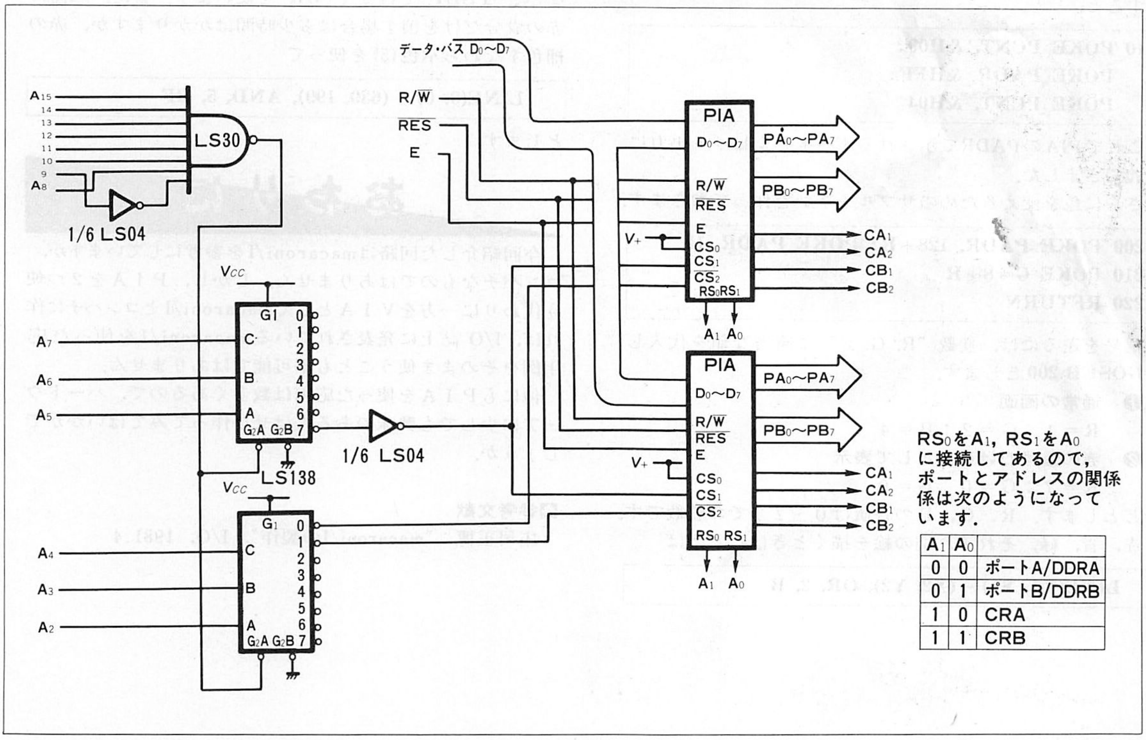
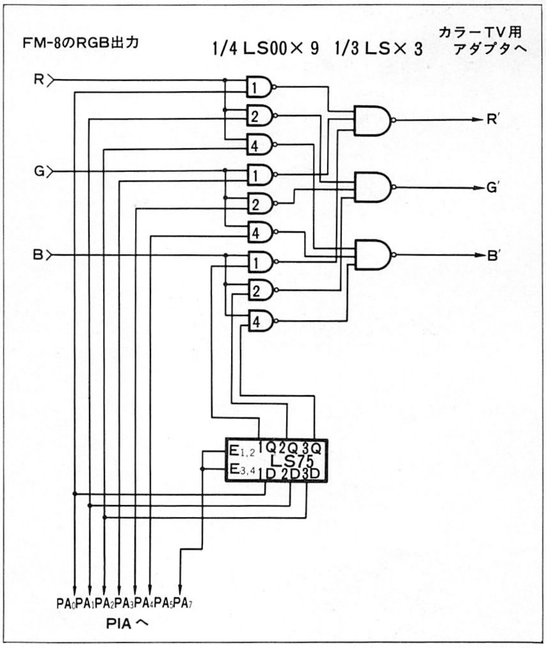


表1　Z80ソフト・カード用コネクタのピン配置

ピン番号	CN8A	CN8B
1	A_0	D_0
2	A_1	D_1
3	A_2	D_2
4	A_3	D_3
5	A_4	D_4　データ・バス
6	A_5	D_5
7	A_6　アドレス・バス	D_6
8	A_7	D_7
9	A_8	ϕ（クロック）
10	A_9	$\overline{\text{RESET}}$
11	A_{10}	$\overline{\text{INT}}$（6809の$\overline{\text{IRQ}}$）
12	A_{11}	$\overline{\text{NMI}}$
13	A_{12}	GND
14	A_{13}	GND　GND
15	A_{14}	GND
16	A_{15}	E
17		Vcc
18	⑱・⑲＋⑰がZ80のWAITへ	Vcc　＋5V
19		Vcc
20	R/$\overline{\text{W}}$	Q

図2　PIAの応用



使用法

PIAの使用法はmacaroni/Iの記事に詳しく出ているので，そちらを参照してください．BASICからPOKE文とPEEK文により制御できるので操作は簡単です．

ここでPIAの簡単な応用を紹介します．FM-8のビデオ出力にはRGB出力とコンポジット出力がありますが，RGB出力のほうはTTLレベルの信号です．

そこで，このRGBの各信号をTTLで制御して，たとえば，R(赤)だけを“白”(RとGとB）として出力したり，あるいは“RをG”，“GをB”，“BをR”と入れ替えて出力したりします．これにより瞬時に画面の色を変えたり，3枚独立の白黒画面にしたりできます．

回路を図2に示しますが，信号線が9本必要なのでラッチを使って2回に分けて信号を送るようにしています．

図に示したように配線した場合，PIAの制御は次のようになります．

まず，PIAを出力に設定します．

PADR＝ポートのアドレス

PCNT＝ポートのコントロール・レジスタのアドレス

とすると，

```
10 POKE PCNT, &H00:
   POKE PADR, &HFF:
   POKE PCNT, &H04
```

これでPIAのPADRで示されるポートを全ビット出力に定義できました．

さらに色を変えるためのサブルーチンを作っておきます．

```
200 POKE PADR, 128+B: POKE PADR, B
210 POKE G*8+R
220 RETURN
```

信号を送るには，変数“R, G, B”に適当な値を代入してGOSUB 200とします．

❶ **通常の画面**
R＝1：G＝2：B＝4

❷ **赤の信号だけを白として表示**
R＝7：G＝0：B＝0

などとします．R，G，Bの各値は0～7までの整数です．

赤，青，緑，それぞれ別の絵を描くときはたとえば

```
LINE(X1, Y1)－(X2, Y2), OR, 2, B
```

などと“PSET”ではなく“OR”を使います．また，画面の赤の成分だけを消す場合は多少時間はかかりますが，赤の補色すなわち水色(5)を使って

```
LINE(0, 0)－(639, 199), AND, 5, BF
```

とします．

おわりに

今回紹介した回路はmacaroni/Iを参考にしていますが，コンパチなものではありません．しかし，ＰＩＡを２つ使う代わりに一方をＶＩＡとしてmacaroni/Iとコンパチに作れば，I/O 誌上に発表されているmacaroni/Iを使った応用例をそのまま使うことも不可能ではありません．

他にもＰＩＡを使った応用は数多くあるので，ハードウェアに少しでも興味のある人はぜひ作ってみてはいかがでしょうか．

参考文献

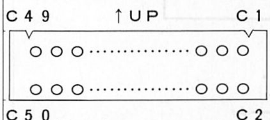
庄司正博：〝macaroni/Iの製作〟, I/O, 1981.4

FM-8の拡張インターフェイス

■安城マイコン同好人

FM-8のシステム拡張ユニットを首を長〜くして待っている人のために，拡張インターフェイスを発表します．

図1．拡張インターフェイス



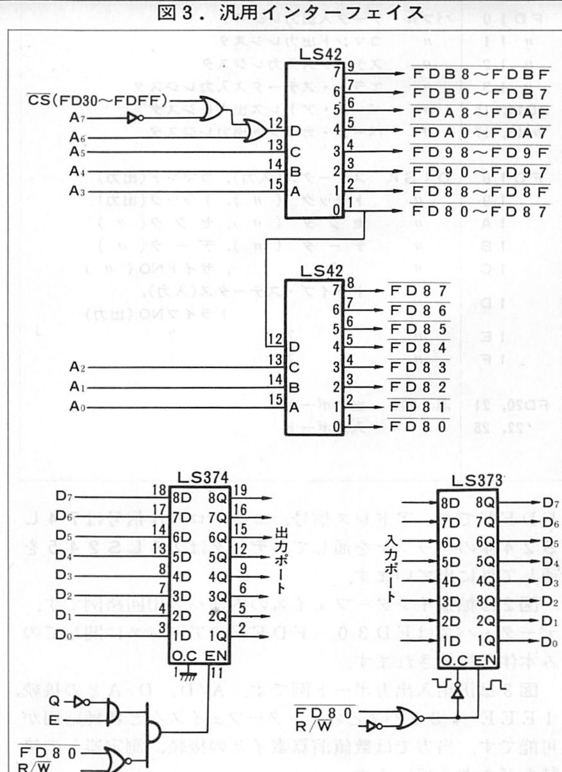
（本体背面より見た図）
空白部：不明

C1	RESET	C2	
C3		C4	$\overline{CS}$
C5	$\overline{IRQ}$	C6	D_0
C7	$\overline{NMI}$	C8	D_1
C9	$\overline{FIRQ}$	C10	D_2
C11	$\overline{HALT}$	C12	D_3
C13	BA	C14	D_4
C15	BS	C16	D_5
C17		C18	D_6
C19		C20	D_7
C21		C22	GND
C23	R/$\overline{W}$	C24	GND
C25	GND	C26	A_0
C27	A_1	C28	A_2
C29	A_3	C30	A_4
C31	A_5	C32	A_6
C33	A_7	C34	A_8
C35	A_9	C36	A_{10}
C37	A_{11}	C38	A_{12}
C39	A_{13}	C40	A_{14}
C41	A_{15}	C42	
C43	MRDY	C44	
C45	＋5V	C46	＋5V
C47	Q	C48	GND
C49	E	C50	GND

$\overline{CS}$：FD00〜FD2F（FM-8内部で使用中または拡張インターフェイスユニットなどで使用予定？ のシステムI/Oエリア，$\overline{CS}$）

EXPANSION PORTについて

図3．汎用インターフェイス



FM-8にはEXPANSION PORTが用意されていますが，各端子にどの信号が出ているのかマニュアルにも載っていません．

システム拡張ユニット内のバスに関しては公表する予定らしいのですが，発売予定のままではどうしようもないのが現状です．そこで，当方で独自に調査しました．まだ不明の部分もありますが，一般的使用に関しては充分実用になると思います．

図1は本体背面に出ているEXPANSION PORTの端子とその信号の対応関係です．この中で，C4の$\overline{CS}$はシステムで使用中またはシステム拡張ユニットなどで使用予定（？）のI/Oポート用チップ・セレクト信号です．

アドレスはFD00〜FD2Fの48ポートです．ですから，ユーザーの使用可能なI/OアドレスはFD30〜

システムI/Oポート

アドレス	内容
FD00	b_0：カセット・データ出力，b_1：カセット・モータ・ON/OFF，b_6：プリンタ STROBE出力，b_7：？
FD01	プリンタへData出力
FD02	b_0：プリンタよりのBUSY入力，b_1，b_3：プリンタ・ステータス，b_7：カセット・データ入力
FD03	b_0：Bell ON/OFF，　b_7：？
FD04	b_1：？ 入力
FD05	b_0：Z80/6809(？)　b_7：サブシステムよりのBUSY信号(Ready)
FD08	音声合成 (出力)
09	〃 (出力)
0A	〃 (入力)
0B	〃 (出力)
FD0C	アナログ・入力ポート (コントロール出力)
FD10	バブル データ入出力レジスタ
〃 11	〃 コマンド出力レジスタ
〃 12	〃 ステータス入力レジスタ
〃 13	〃 エラー・ステータス入力レジスタ
FD14, 15	〃 ページ・アドレス出力レジスタ
〃 16, 17	〃 ページ・カウント出力レジスタ
FD18	DISK ステータス(入力)，コマンド(出力)
19	〃 トラック (〃)，トラック(出力)
1A	〃 セクタ (〃)，セクタ(〃)
1B	〃 データ (〃)，データ(〃)
1C	〃 ？ ，サイドNO.(〃)
1D	〃 ドライブ・ステータス(入力)，ドライブNO.(出力)
1E	〃 ？ ？
1F	〃 ？ ？
FD20, 21	漢字 出力ポート
22, 23	〃 入力ポート

FDFFです．アドレス信号，コントロール信号は74LS244のバッファを通して，データは74LS245を通して外に出ています．

図2は拡張インターフェイスの内部バス用回路例です．データ・バスはFD30～FDFFのアドレスに関してのみ本体と接続されます．

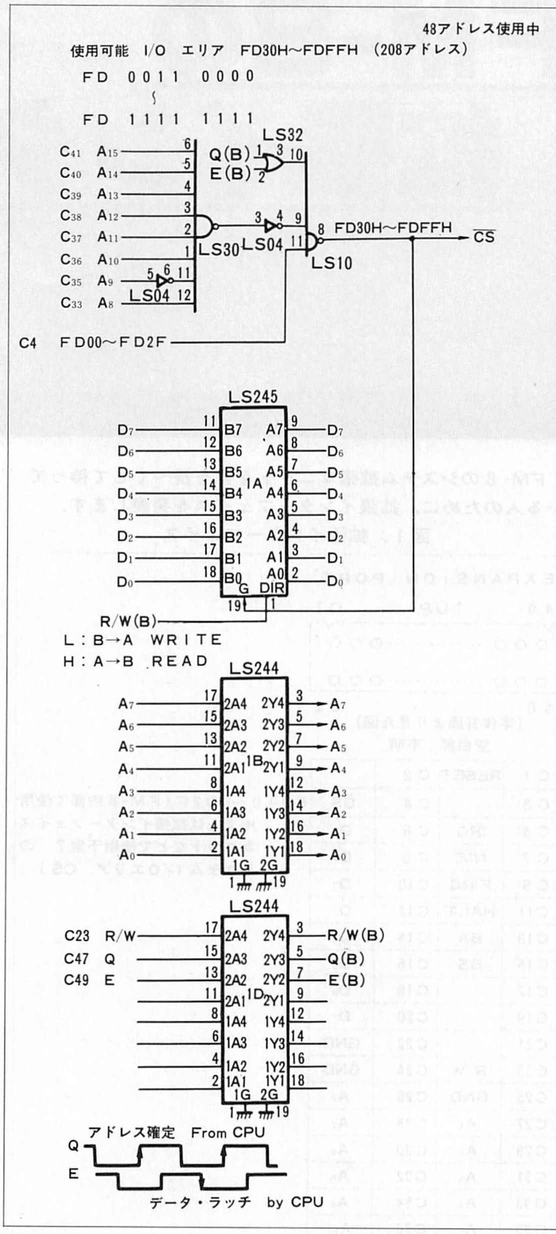
図3は汎用入出力ポート例です．A/D，D/Aとの接続，IEEE-488バスとのインターフェイスなど各種応用が可能です．当方では数値演算素子との接続，測定器との接続などを考えています．

現在，出力ポートからデータ・アドレスを出し，入力ポートからデータを読み取るタイプのROM内容読み取り器を構成しています．これを使って，IPLROMの内容を読み取り，解読したので次の機会には発表したいと思います．

システム・I/Oポート

BIOSを解読した結果解かったI/Oポートの内容です．誤りもあると思いますので，気付いた人は投稿してください．

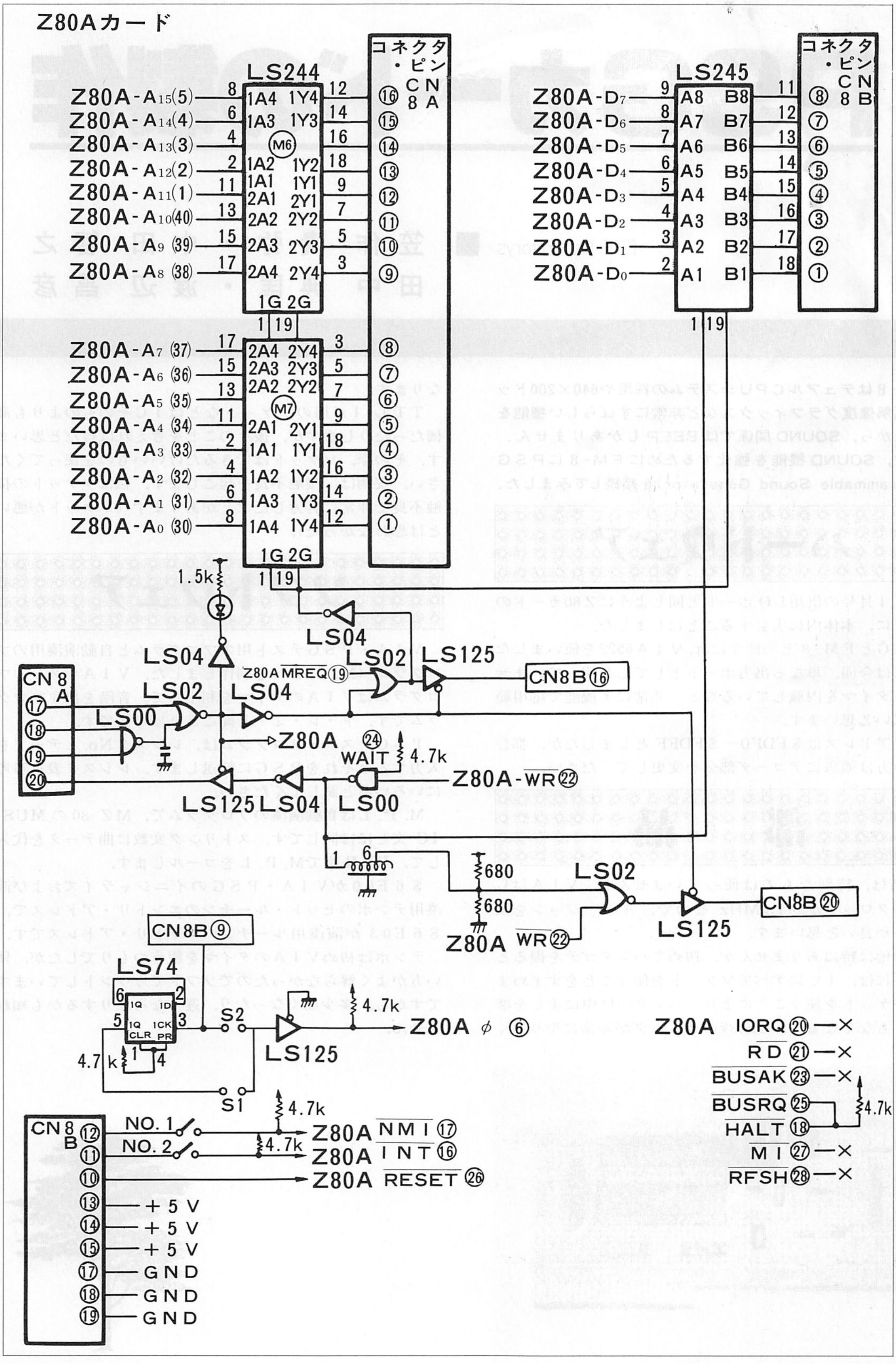
図2．インターフェイス



Z80Aカード

本体とは20ピンのコネクタ2つ(CN8AとCN8B)で接続されています．カード上でZ80の信号から6809システム用R/W，Q，Eを作っています．CN8Bの9ピンから2.5MHzのクロックが供給されています．ジャンパ・ピンにより1.25MHzのクロックにすることも可能です．

自作Ｚ８０Ａカード


Z80Aカード
LS244
LS245
コネクタ・ピン CN 8 A
コネクタ・ピン CN 8 B
Z80A-A15(5)
Z80A-A14(4)
Z80A-A13(3)
Z80A-A12(2)
Z80A-A11(1)
Z80A-A10(40)
Z80A-A9 (39)
Z80A-A8 (38)
Z80A-A7 (37)
Z80A-A6 (36)
Z80A-A5 (35)
Z80A-A4 (34)
Z80A-A3 (33)
Z80A-A2 (32)
Z80A-A1 (31)
Z80A-A0 (30)
Z80A-D7
Z80A-D6
Z80A-D5
Z80A-D4
Z80A-D3
Z80A-D2
Z80A-D1
Z80A-D0
M6
M7
1G 2G
1.5k
LS04
LS02
LS125
Z80A MREQ ⑲
CN8B⑯
CN 8 A
LS00
Z80A ㉔
WAIT 4.7k
Z80A-WR㉒
LS125 LS04 LS00
680
680
Z80A WR㉒
CN8B⑳
CN8B⑨
LS74
S2
S1
4.7k
Z80A φ ⑥
Z80A IORQ⑳—×
RD㉑—×
BUSAK㉓—×
BUSRQ㉕
HALT⑱
MI㉗—×
RFSH㉘—×
CN 8 B
NO. 1
NO. 2
Z80A NMI ⑰
Z80A INT ⑯
Z80A RESET ㉖
+5V
+5V
+5V
GND
GND
GND

PSGカードの製作

ETF Laboratorys ■ 笠作貴弥・中田智之
田中典匡・渡辺昌彦

FM-8はデュアルCPUシステムの採用や640×200ドットの高解像度グラフィックスなど非常にすばらしい機能を持ちながら，SOUND関係ではBEEPしかありません．そこで，SOUND機能を強化するためにFM-8にPSG(Programmable Sound Generator)を接続してみました．

ハードウェア

I/O 4月号の汎用I/Oポートと同じようにZ80カードの代わりに，本体内に実装することにしました．

ＰＳＧとFM-8との仲介には，ＶＩＡ6522を使いました．ＶＩＡは今回，単なる出力ポートとしてしか使っていませんが，タイマを内蔵しているなど，非常に多機能で応用範囲は広いと思います．

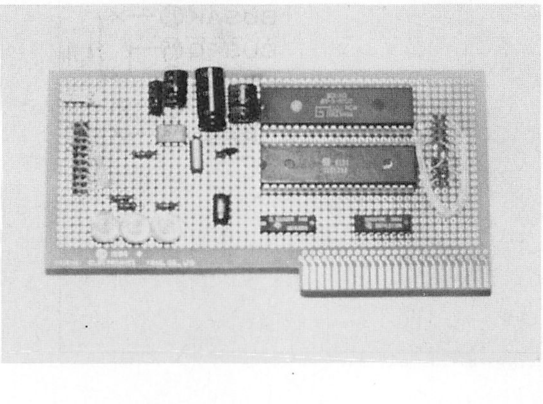
I/Oアドレスは＄FDF0～＄FDFFとしましたが，都合の悪い方は適当にデコーダ部分を変更してください．

部品

部品は，特別なものは使っていませんが，ＶＩＡはＣＰＵのクロックが約1.3MHz なので，Ａバージョンを使った方が良いと思います．

その他は特にありませんが，初めてハンダゴテを握るという方には，ＩＣにすべてソケットを使うことをすすめます．ソケットを使うことにより，ハンダづけ中にＩＣを壊すこともなくなるし，配線後のチェックが非常にやり易くなります．

ＴＴＬ-ＩＣ用のソケットなどはＩＣそのものよりも高価だったりしますが，後々のことを考えれば得だと思います．その際，ソケットはできるだけいいものを使ってください．安物は，接触不良を起こします．私はソケットの接触不良で非常に苦労したことがあります（ソケットが悪いとは思わなかった）．



ソフトウェア

ＶＩＡ・ＰＳＧテスト用のプログラムと自動演奏用のプログラム（M.P.L）を制作しました．ＶＩＡテスト・プログラムはＶＩＡのタイマを利用して，音階を出すプログラムです．ド・レ・ミ…と聞こえればＯＫです．

ＰＳＧテスト・プログラムは，レジスタNo.とデータを入力するとそれをＰＳＧに転送します．レジスタ表を参考にいろいろと試してください．

M.P.Lは自動演奏のプログラムで，ＭＺ-80の**MUSIC**文とほぼ同じです．ストリング変数に曲データを代入して，**USR**文でM.P.Lをコールします．

＄６Ｅ00がＶＩＡ・ＰＳＧのイニシャライズおよび演奏用テンポのセット・ルーチンのエントリ・アドレスで，＄６Ｅ03が演奏用ルーチンのエントリ・アドレスです．

テンポは初めＶＩＡのタイマを使うつもりでしたが，使い方がよく解らなかったのでソフトでカウントしています．ですから，多少速くなったり，遅くなったりするかも知れません．

M.P.Lの文法

1つの音符はオクターブ指定，音形指定，音長指定，音程指定によって構成されます．このうち前の音符と等しい場合，音程データを除いて省略することができます．

❶オクターブ指定（L･M･H）
音階レンジの切り換えです．

❷音形指定（P1～P4）
同じ音長でも，スタッカート気味とか，テヌート気味とかの変化をつけます．

❸音長指定（0～9）
音符の長さを指定します．

❹音程指定（CDEFGABC）（♯）（R）
音程を指定します．半音は♯のみ有効で音程指定の次に♯を書きます．またRのときは，休符となります．

① オクターブ指定

コード	L	M	H
オクターブ	低	中	高

② 音形指定

コード	P1	P2	P3	P4
長　さ	25%	50%	75%	100%

③ 音長指定

コード	0	1	2	3	4	5	6	7	8	9
音　符	𝅘𝅥𝅰	𝅘𝅥𝅯	𝅘𝅥𝅯.	♪	♪.	♩	♩.	𝅗𝅥	𝅗𝅥.	𝅝

④ 音程指定

コード	C	D	E	F	G	A	B	♯	R
音　程	ド	レ	ミ	ファ	ソ	ラ	シ	半音上げる	休　符

あとがき

土曜日の午後に秋葉原に行き，日曜日の夕方にはハード，ソフトを完成させることができました．部品数も少ないので，みなさんぜひ製作してください．

今回のものは応用ソフトも少なく，自動演奏も単音のみでしたが，ＰＳＧの特徴を生かしてポリフォニックで，ダイナミック？なサウンドを作るソフトを制作したいと思います．

◆参考文献

1）I/O'79 9月号"プログラマブル・サウンド・ジェネレータの製作"
2）I/O'81 8月号"レベル3にＰＳＧを接続する"
3）I/O'81 12月号"ＰＳＧによる音楽自動演湊"
4）I/O'81 4月号～"macaroni Ⅰ"

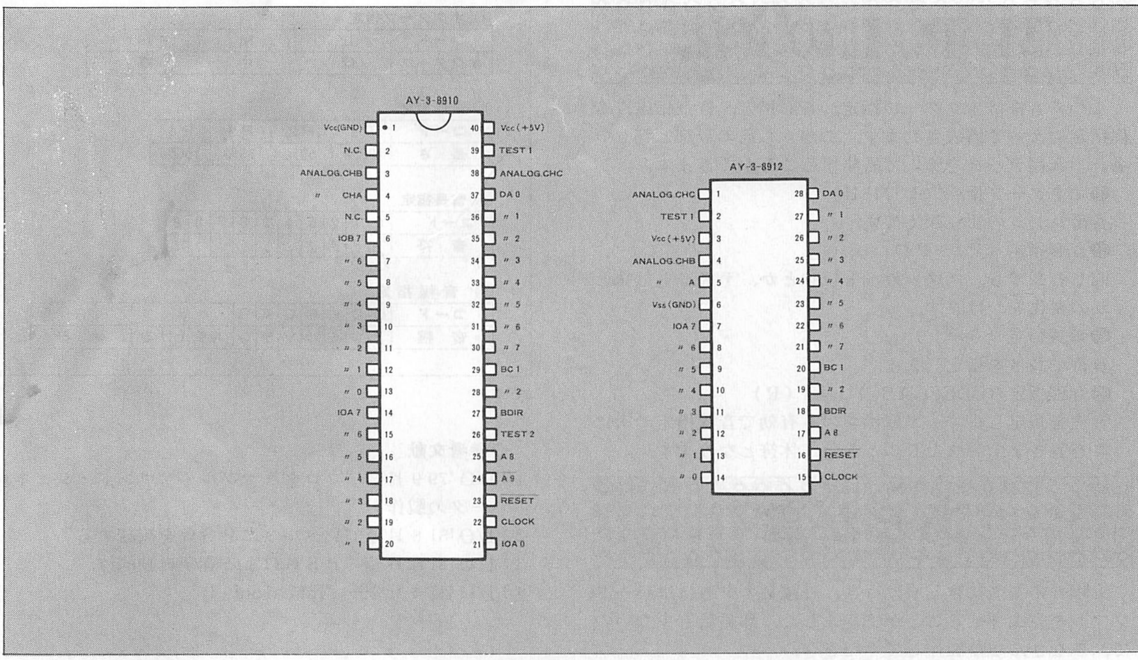
部品表

部品名	形　番	個数
IC	AY-3-8910	1
	6522A	1
	74LS30	1
	74LS138	1
	LM386N	1
R	1.2K	1
	2.2K	1
	10K	2
	20K	1
半固定VR	10K	3
C	1μ 16V（電解）	
	10μ 16V（〃）	
	470μ 16V（〃）	
	4.7μ 16V（タンタル）	1
	0.01μ（マイラ）	1
コネクタ	SP用	1
	20P	2
ソケット	40P	2
基板		1
その他	配線材料など	

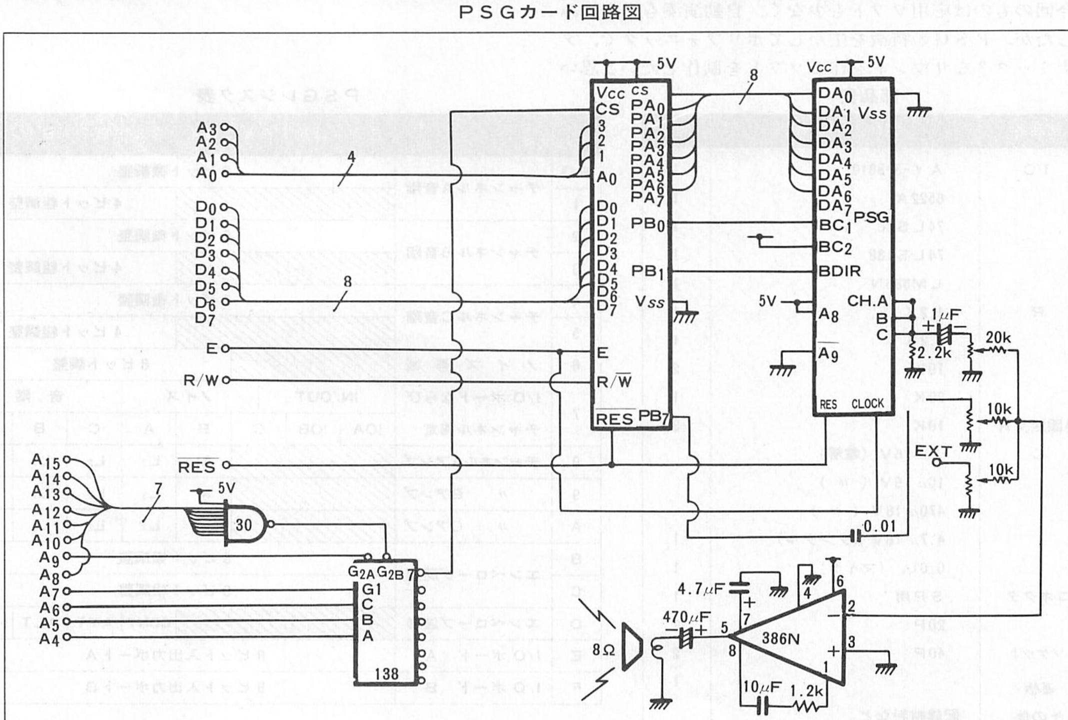
ＰＳＧレジスタ表

レジスタ ＼ ビット		B7	B6	B5	B4	B3	B2	B1	B0
R.0	チャンネルA音階	8ビット微調整							
1						4ビット粗調整			
2	チャンネルB音階	8ビット微調整							
3						4ビット粗調整			
4	チャンネルC音階	8ビット微調整							
5						4ビット粗調整			
6	ノイズ帯域				8ビット調整				
7	I/Oボードならび	IN/OUT		ノイズ			音階		
	チャンネル指定	IOA	IOB	C	B	A	C	B	A
8	チャンネルAアンプ				M	L_3	L_2	L_1	L_0
9	〃 Bアンプ				M	L_3	L_2	L_1	L_0
A	〃 Cアンプ				M	L_3	L_2	L_1	L_0
B	エンベロープ周期	8ビット微調整							
C		8ビット粗調整							
D	エンベロープ波形					CONT	ATT	ALT	HOLD
E	I/Oボード A	8ビット入出力ポートA							
F	I/Oボード B	8ビット入出力ポートB							

AＹ-3のピン配置図



ＰＳＧカード回路図



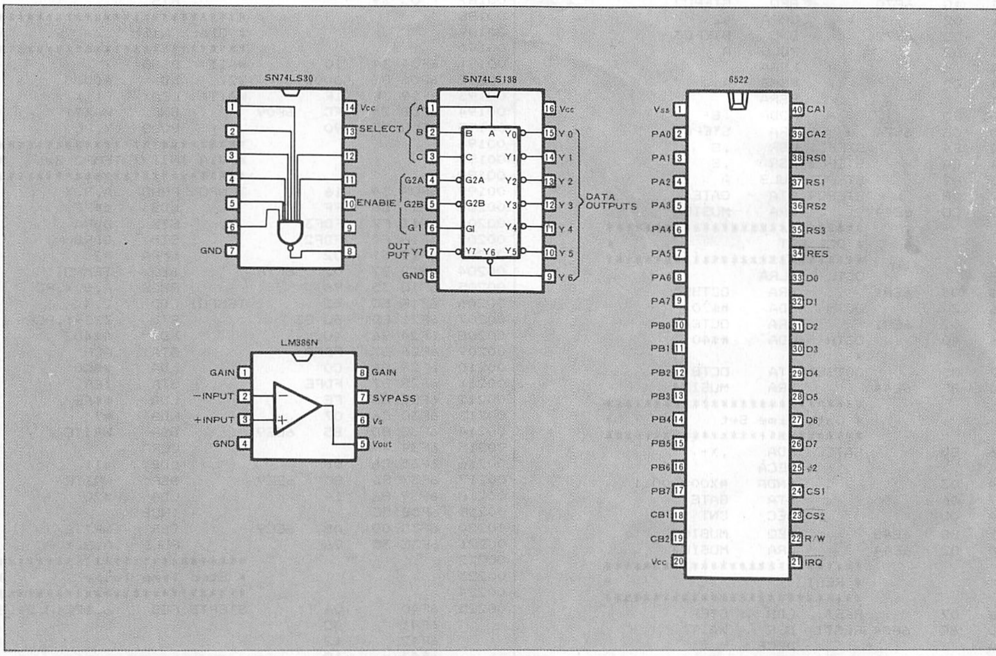
サンプル・プログラム

```
10 ' DEMO PROGRAM
20 CLEAR 300,&H6DFF
30 DEFUSR0=&H6E00:DEFUSR1=&H6E03
40 A%=USR0(850) ' TEMPO SET & INIT
50 READ A$:A$=USR1(A$)
60 READ A$:A$=USR1(A$)
70 DATA P25MDE9F3RFGFE1ED3CLAMP49DP17D
80 DATA P25MDE9F3RFGFE1ED3CDP49AP17A
```

ＰＳＧテスト・プログラム

```
10 ' PSG TEST PROGRAM
20 VORB=&HFDF0:VORA=&HFDF1
30 DDRB=&HFDF2:DDRA=&HFDF3:PCR=&HFDFC
40 POKE PCR,0:POKE DDRA,255:POKE DDRB,255
50 INPUT"REG,DATA";REG,DAT:GOSUB 60:GOTO 50
60 '  DATA SET ROUTINE
70 POKE VORB,3:POKE VORA,REG:POKE VORB,0
80 POKE VORA,DAT:POKE VORB,2:POKE VORB,0
90 RETURN
```

ピン配置図



V I A　テスト・プログラム

```
10 ' VIA TEST PROGRAM
20 VIADDRB=&HFDF2:VIAACR=&HFDFB
30 VIAT1L=&HFDF4:VIAT1H=&HFDF5
40 POKE VIADDRB,&HB0
50 POKE VIAACR,&HC0
60 RESTORE
70 FOR I=1 TO 8:READ C
80 C=INT(618750!/C)
90 POKE VIAT1L,C MOD 256
100 POKE VIAT1H,INT(C/256)
110 FOR J=1 TO 500:NEXT:NEXT:GOTO60
120 DATA 261.3,293.66,329.63,349.23,392,440,493.88,523.25
```

P S G　コントロール・プログラム・リスト

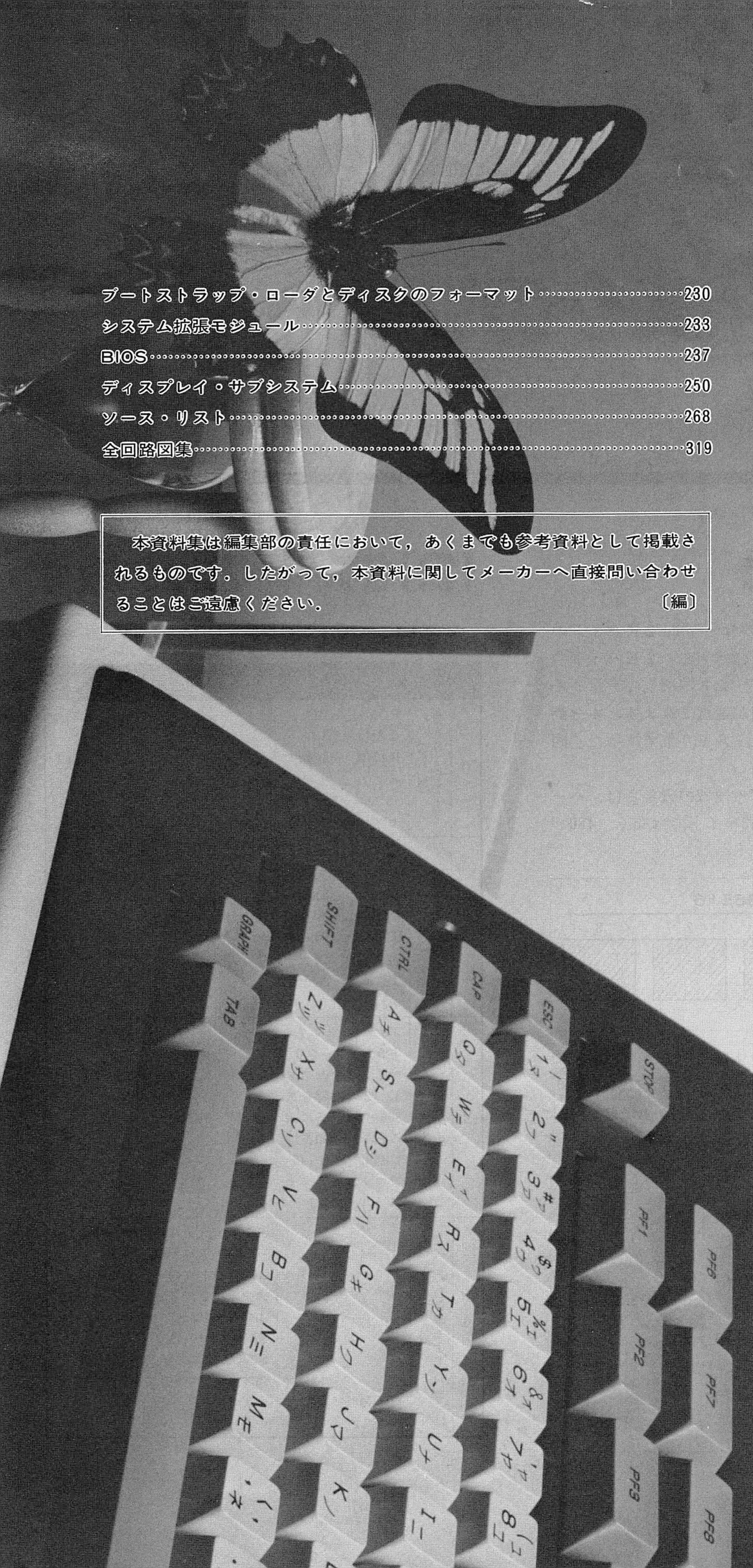
```
                         NAM    M.P.L
                         ***************************
                         * PSG Contoroll Program.  *
                         *    for FM-8 Version 1.2 *
                         * (C) ETF Laboratory.     *
                         *   Written by T.Kasasaku.*
                         ***************************
              6E00       BEGIN  EQU    $6E00
              006E              SETDP  BEGIN/256
6E00                            ORG    BEGIN
                                OPT    S
                         ***************************
                         * VIA I/O Address         *
                         ***************************
              FDF1       ORA    EQU    $FDF1
              FDF0       ORB    EQU    $FDF0
              FDF3       DDRA   EQU    $FDF3
              FDF2       DDRB   EQU    $FDF2
              FDF4       T1L    EQU    $FDF4
              FDFB       ACR    EQU    $FDFB
              FDFD       IFR    EQU    $FDFD
              FDFE       IER    EQU    $FDFE
                         ***************************
                         * Program ENTRY           *
                         ***************************
6E00 16       010C 6F0F         LBRA   TEMPO   INIT Routine
6E03 20       07   6E0C         BRA    MUSIC   Play Routine
                         ***************************
                         * Wrok Area               *
                         ***************************
6E05          0001       OCTB   RMB    1
6E06          0001       GATE   RMB    1
6E07          0001       STEP   RMB    1
6E08          0001       GATE1  RMB    1
6E09          0001       STEP1  RMB    1
6E0A          0001       GATE2  RMB    1
6E0B          0001       CNT    RMB    1
                         ***************************
                         * MUSIC PLAY              *
                         ***************************
6E0C 34       7E         MUSIC  PSHS   A,B,DP,X,Y,U
6E0E 81       03                CMPA   #3
6E10 27       02   6E14         BEQ    MUSIC2
6E12 35       FE         MUSIC1 PULS   A,B,DP,X,Y,U,PC
6E14 1F       50         MUSIC2 TFR    PC,D
6E16 1F       8B                TFR    A,DP
6E18 A6       84                LDA    ,X
6E1A 27       F6   6E12         BEQ    MUSIC1
6E1C 97       0B                STA    CNT
6E1E AE       01                LDX    1,X
6E20 A6       80         MUSIC3 LDA    ,X+
6E22 81       30                CMPA   #'0
6E24 25       1E   6E44         BCS    MUSIC4
6E26 81       40                CMPA   #'@
6E28 25       20   6E4A         BCS    STEPC
6E2A 81       41                CMPA   #'A
6E2C 25       16   6E44         BCS    MUSIC4
6E2E 81       48                CMPA   #'H
6E30 27       4D   6E7F         BEQ    OCTH
6E32 25       67   6E9B         BCS    PLAY
6E34 81       4C                CMPA   #'L
6E36 27       40   6E78         BEQ    OCTL
6E38 81       4D                CMPA   #'M
6E3A 27       3F   6E7B         BEQ    OCTM
6E3C 81       50                CMPA   #'P
6E3E 27       45   6E85         BEQ    GATEC
6E40 81       52                CMPA   #'R
6E42 27       4E   6E92         BEQ    REST
6E44 0A       0B         MUSIC4 DEC    CNT
6E46 26       D8   6E20         BNE    MUSIC3
6E48 35       FE         MUSIC5 PULS   A,B,DP,X,Y,U,PC
                         ***************************
                         * Step Time Set           *
                         ***************************
6E4A 84       0F         STEPC  ANDA   #%00001111
6E4C 31       8D 00F0           LEAY   STEPTB,PCR
6E50 31       A6                LEAY   A,Y
6E52 A6       A4                LDA    ,Y
6E54 97       07                STA    STEP
6E56 34       02                PSHS   A
6E58 96       06                LDA    GATE
6E5A 27       12   6E6E         BEQ    STEPC1
```

PSG コントロール・プログラム・リスト

```
6E5C 81    01               CMPA   #1
6E5E 27    10    6E70       BEQ    STEPC2
6E60 81    02               CMPA   #2
6E62 26    0E    6E72       BNE    STEPC3
6E64 35    02               PULS   A
6E66 44                     LSRA
6E67 34    02               PSHS   A
6E69 44                     LSRA
6E6A AB    E0               ADDA   ,S+
6E6C 20    06    6E74       BRA    STEPC4
6E6E 64    E4         STEPC1 LSR   ,S
6E70 64    E4         STEPC2 LSR   ,S
6E72 35    02         STEPC3 PULS  A
6E74 97    08         STEPC4 STA   GATE1
6E76 20    CC    6E44       BRA    MUSIC4
                      **************************
                      * OCT SET                *
                      **************************
6E78 4F               OCTL   CLRA
6E79 20    06    6E81        BRA   OCTSET
6E7B 86    20         OCTM   LDA   #$20
6E7D 20    02    6E81        BRA   OCTSET
6E7F 86    40         OCTH   LDA   #$40
                      *
6E81 97    05         OCTSET STA   OCTB
6E83 20    BF    6E44        BRA   MUSIC4
                      **************************
                      * Gate Time Set          *
                      **************************
6E85 A6    80         GATEC  LDA   ,X+
6E87 4A                      DECA
6E88 84    03                ANDA  #%00000011
6E8A 97    06                STA   GATE
6E8C 0A    0B                DEC   CNT
6E8E 27    B8    6E48        BEQ   MUSIC5
6E90 20    B2    6E44        BRA   MUSIC4
                      **************************
                      * REST                   *
                      **************************
6E92 D6    07         REST   LDB   STEP
6E94 8D    6E    6F04 REST1  BSR   WAIT
6E96 5A                      DECB
6E97 26    FB    6E94        BNE   REST1
6E99 20    A9    6E44        BRA   MUSIC4
                      **************************
                      * PLAY 1-Note            *
                      **************************
6E9B D6    08         PLAY   LDB   GATE1
6E9D D7    0A                STB   GATE2
6E9F D6    07                LDB   STEP
6EA1 D7    09                STB   STEP1
6EA3 4A                      DECA
6EA4 84    07                ANDA  #%00000111
6EA6 48                      ASLA
6EA7 48                      ASLA
6EA8 9B    05                ADDA  OCTB
6EAA E6    84                LDB   ,X
6EAC C1    23                CMPB  #'#
6EAE 26    06    6EB6        BNE   PLAY1
6EB0 8B    02                ADDA  #2
6EB2 30    01                LEAX  1,X
6EB4 0A    0B                DEC   CNT
6EB6 31    8D 0090    PLAY1  LEAY  NOTETB,PCR
6EBA 31    A6                LEAY  A,Y
6EBC A6    A0                LDA   ,Y+
6EBE 5F                      CLRB
6EBF 8D    28    6EE9        BSR   WRITE
6EC1 A6    A4                LDA   ,Y
6EC3 5C                      INCB
6EC4 8D    23    6EE9        BSR   WRITE
6EC6 86    0F                LDA   #15
6EC8 C6    08                LDB   #8
6ECA 8D    1D    6EE9        BSR   WRITE
6ECC 8D    36    6F04 PLAY2  BSR   WAIT
6ECE 0D    0A                TST   GATE2
6ED0 27    06    6ED8        BEQ   PLAY3
6ED2 0A    0A                DEC   GATE2
6ED4 26    02    6ED8        BNE   PLAY3
6ED6 8D    07    6EDF        BSR   ENVSET
6ED8 0A    09         PLAY3  DEC   STEP1
6EDA 26    F0    6ECC        BNE   PLAY2
6EDC 16    FF65 6E44         LBRA  MUSIC4
                      **************************
                      * RELEASE SET            *
                      **************************
6EDF 86    10         ENVSET LDA   #$10
6EE1 C6    08                LDB   #8
6EE3 8D    04    6EE9        BSR   WRITE
6EE5 86    09                LDA   #9
6EE7 C6    0D                LDB   #13
                      **************************
                      * Write to PSG           *
                      *    A=Data  B=Reg.No.   *
                      **************************
6EE9 34    02         WRITE  PSHS  A
6EEB 86    03                LDA   #3
6EED B7    FDF0              STA   ORB
6EF0 F7    FDF1              STB   ORA
6EF3 7F    FDF0              CLR   ORB
6EF6 35    02                PULS  A
6EF8 B7    FDF1              STA   ORA
6EFB 86    02                LDA   #2
6EFD B7    FDF0              STA   ORB
6F00 7F    FDF0              CLR   ORB
6F03 39                      RTS
                      **************************
                      * Clock Wait             *
                      **************************
6F04 34    10         WAIT   PSHS  X
6F06 8E    0000       ZZZ    LDX   #0000
6F09 30    1F         WAIT1  LEAX  -1,X
6F0B 26    FC    6F09        BNE   WAIT1
6F0D 35    90                PULS  X,PC
                      **************************
                      * VIA INIT / TEMPO Set   *
                      **************************
6F0F 34    16         TEMPO  PSHS  A,B,X
6F11 C6    FF                LDB   #$FF
6F13 F7    FDF3              STB   DDRA
6F16 F7    FDF2              STB   DDRB
6F19 81    02                CMPA  #2
6F1B 27    02    6F1F        BEQ   TEMPO1
6F1D 35    96                PULS  A,B,X,PC
6F1F EC    02         TEMPO1 LDD   2,X
6F21 ED    8C E3             STD   ZZZ+1,PCR
6F24 86    40                LDA   #$40
6F26 B7    FDFB              STA   ACR
6F29 86    C0                LDA   #$C0
6F2B B7    FDFE              STA   IER
6F2E 86    FE                LDA   #$FE
6F30 C6    07                LDB   #7
6F32 8D    B5    6EE9        BSR   WRITE
6F34 4F                      CLRA
6F35 C6    0B                LDB   #11
6F37 8D    B0    6EE9        BSR   WRITE
6F39 86    14                LDA   #20
6F3B 5C                      INCB
6F3C 8D    AB    6EE9        BSR   WRITE
6F3E 35    96                PULS  A,B,X,PC
                      **************************
                      * Step Time Table        *
                      **************************
6F40       06         STEPTB FCB   6,12,18,24,36
6F41       0C
6F42       12
6F43       18
6F44       24
6F45       30                FCB   48,72,96,144,192
6F46       48
6F47       60
6F48       90
6F49       C0
                      **************************
                      * Note Table             *
                      **************************
6F4A       AE00       NOTETB FDB   $AE00,$A400,$9B00,$9200
6F4C       A400
6F4E       9B00
6F50       9200
6F52       2501              FDB   $2501,$1501,$0501,$F600
6F54       1501
6F56       0501
6F58       F600
6F5A       E800              FDB   $E800,$DB00,$DB00,$CD00
6F5C       DB00
6F5E       DB00
6F60       CD00
6F62       C300              FDB   $C300,$B800,0,0
6F64       B800
6F66       0000
6F68       0000
6F6A       5700              FDB   $5700,$5200,$4D00,$4900
6F6C       5200
6F6E       4D00
6F70       4900
6F72       9200              FDB   $9200,$8A00,$8200,$7B00
6F74       8A00
6F76       8200
6F78       7B00
6F7A       7400              FDB   $7400,$6D00,$6D00,$6600
6F7C       6D00
6F7E       6D00
6F80       6600
6F82       6100              FDB   $6100,$5C00,0,0
6F84       5C00
6F86       0000
6F88       0000
6F8A       2B00              FDB   $2B00,$2900,$2600,$2400
6F8C       2900
6F8E       2600
6F90       2400
6F92       4900              FDB   $4900,$4500,$4100,$3D00
6F94       4500
6F96       4100
6F98       3D00
6F9A       3A00              FDB   $3A00,$3600,$3600,$3300
6F9C       3600
6F9E       3600
6FA0       3300
6FA2       3000              FDB   $3000,$2E00,0,0
6FA4       2E00
6FA6       0000
6FA8       0000
                             END
```

資料編

本資料集は編集部の責任において，あくまでも参考資料として掲載されるものです．したがって，本資料に関してメーカーへ直接問い合わせることはご遠慮ください．〔編〕

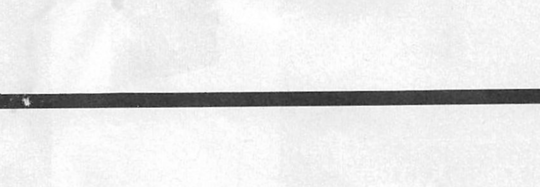


ブートストラップ・ローダとディスクのフォーマット

〔1〕はじめに

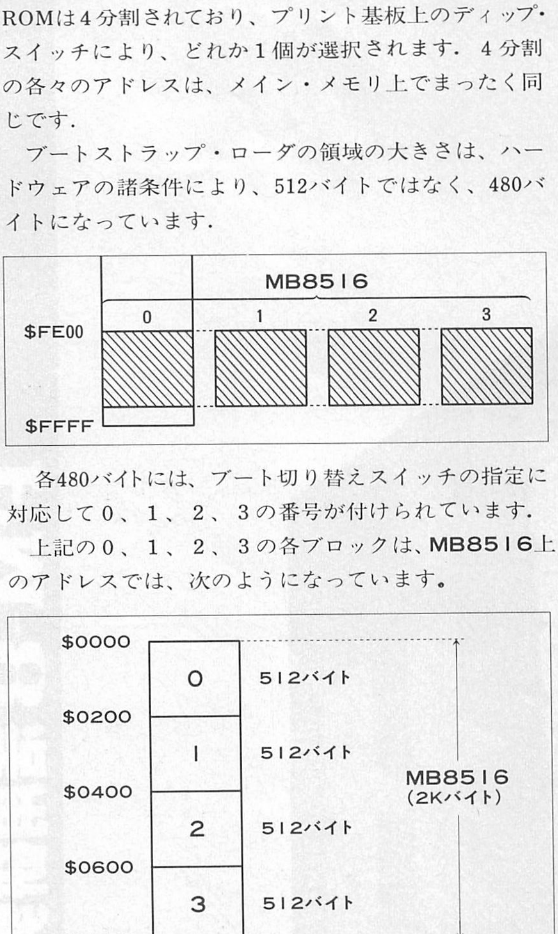
FM-8のブートストラップ・ローダは2Kバイトの EPROM MB8516に格納されます. この2Kバイトの ROMは4分割されており、プリント基板上のディップ・スイッチにより、どれか1個が選択されます. 4分割の各々のアドレスは、メイン・メモリ上でまったく同じです.

ブートストラップ・ローダの領域の大きさは、ハードウェアの諸条件により、512バイトではなく、480バイトになっています.



各480バイトには、ブート切り替えスイッチの指定に対応して0、1、2、3の番号が付けられています.

上記の0、1、2、3の各ブロックは、MB8516上のアドレスでは、次のようになっています。



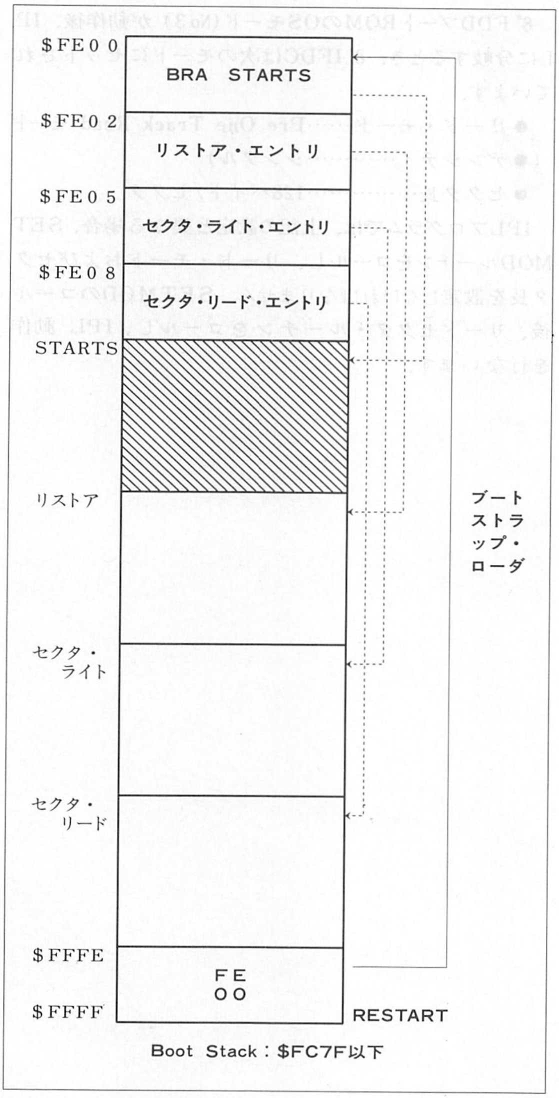
〔2〕各ブート・ブロックの機能

各ブート・ブロックの機能は以下のとおりです。

		5″サポート版	8″サポート版	備　考
0	BASICモード (ROM)	ROM BASIC または、disk BASIC を起動する. (5″フロッピーアクセス・ルーチン).	同　左	ブート・スタック $FC7F以下
1	バブル・モード (RAM)	バブル・ユニット 0の先頭の16ページの内容を$300～$4FFに読み込み、$300番地に制御を渡す(5″フロッピーアクセス・ルーチン).	同　左	ブート・スタック $FC7F以下
2	DOSモード (RAM)	5″ディスク・ユニット0、トラック0、サイド0のセクタ1、2の内容を$300～$4FFに読み込み、$300番地に制御を渡す (5″フロッピーアクセス・ルーチン)	同　左	ブート・スタック $FC7F
3	DOSモード (RAM)	(ROM BASIC デバッグ用)	8″ディスク・ユニット0、トラック0、サイド0のセクタ1、2、3、4の内容を$300～$4FFに読み込み、$300番地に制御を渡す (8″フロッピーアクセス・ルーチン)	ブート・スタック $02EF以下

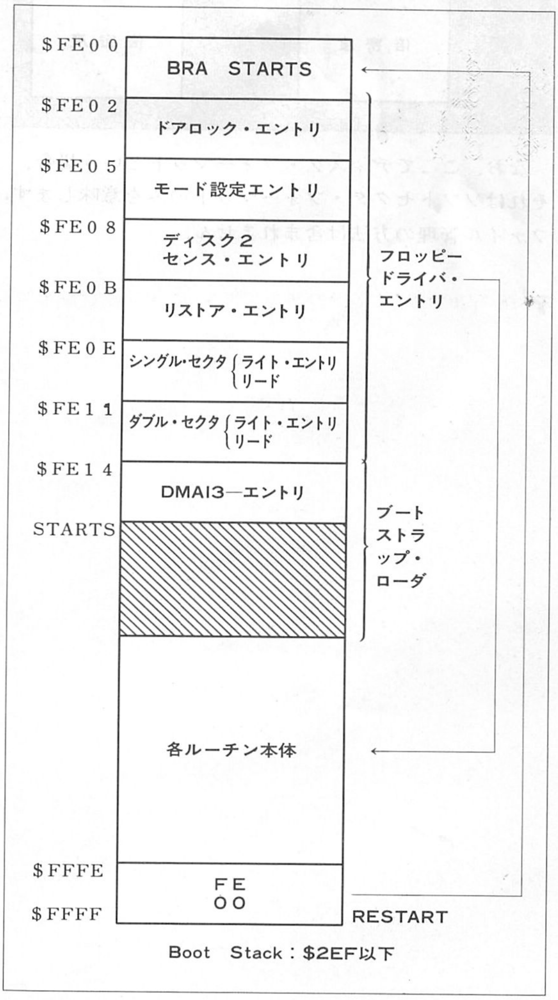
[3]5″版ブートROM構成メモ

5″版のブートROMは次のような構成になっています.



[4]8″版ブートROM構成メモ

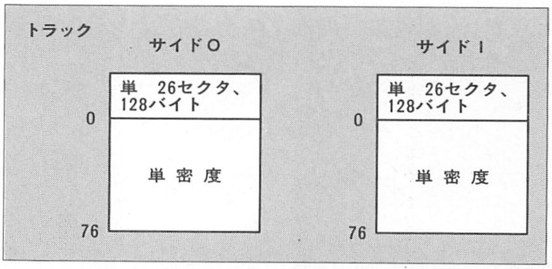
8″版のブートROMブート・ブロックNo.3の構成は次のとおりです.



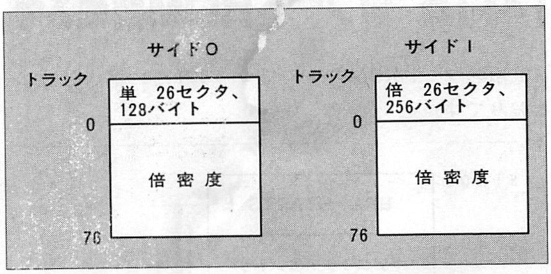
[5]ディスクのフォーマット規定

8″フロッピーディスクのフォーマットはIBMのディスク・フォーマットと同じにする．トラック0はこれにより次のように定まります．

(1)単密度



(2)倍密度



なお、ここでディスク・フォーマットという場合、それはソフトセクタ・フォーマットのみを意味します．ファイル管理の方法は含まれません．

[6] 8″ブートROMのOSモードについて

8″FDDブートROMのOSモード(No.3)が動作後、IPLに分岐するとき、8″IFDCは次のモードにセットされています．

- ●リード・モード……Pre One Track Read モード
- (●デンシテイ…………シングル)
- ●セクタ長……………128バイト/セクタ

IPLプログラムでは、上記の設定と異なる場合、SETMODルーチンをコールし、リード・モードおよびセクタ長を設定しなければなりません．SETMODのコール後、リードセクタ・ルーチンをコールし、IPL 動作を行ないます．

システム拡張モジュール

[1]適　用

FM-8システム拡張ユニットMB26001内に組み込むシステム拡張モジュールMB22201について解説します.

[2]概　要

FM-8システム拡張モジュールはFM-8 本体の機能の他にさらに新しい機能を付加するためのシステム・アップの可能性を提供する各種モジュールと、FM-8本体とをインターフェイスするモジュールであり、FM-8本体とは50ピンのフラット・ケーブルで接続され、各種モジュールとはFACOM L-8Aマザーバスで接続されます.

図１に拡張システムの構成図を示します.

FM-8システム拡張モジュールは以下の回路で構成されます.

①アドレス・バッファ回路
②データ・バッファ回路
③DMA制御回路
④ミニフロッピーインターフェイス回路
⑤標準フロッピーインターフェイス回路

図２にシステム拡張モジュールのブロック構成図を示します.

図１　拡張システム構成図

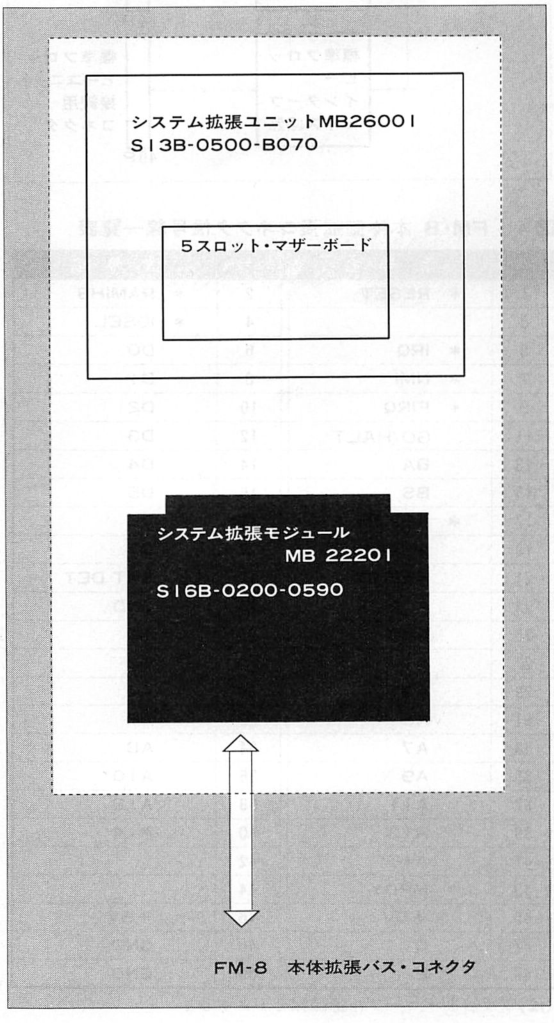


図2　システム拡張モジュール・ブロック構成図

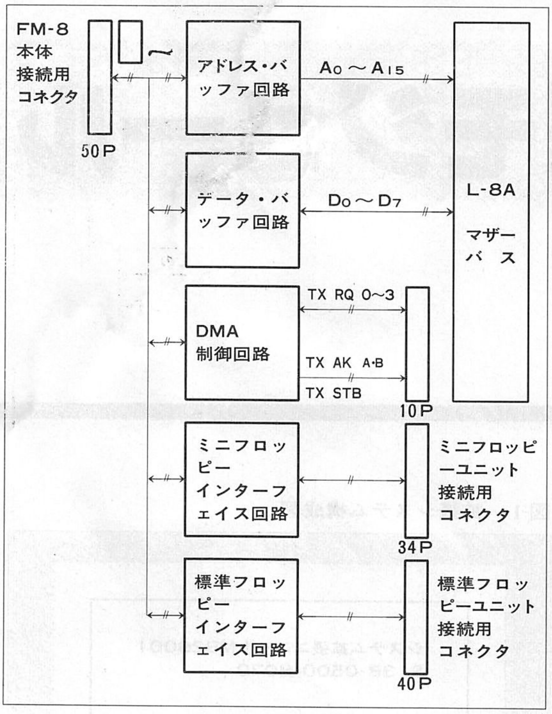


図4　FM-8 本体側拡張コネクタ信号線一覧表

コネクタNo.	信号名	コネクタNo.	信号名
1	* RESET	2	* RAMIHB
3		4	* IOSEL
5	* IRQ	6	D0
7	* NMI	8	D1
9	* FIRQ	10	D2
11	GO/HALT	12	D3
13	BA	14	D4
15	BS	16	D5
17	* REF REQ X	18	D6
19	REFGRNT	20	D7
21	REF CK	22	* EXT DET
23	R/W	24	GND
25	GND	26	A0
27	A1	28	A2
29	A3	30	A4
31	A5	32	A6
33	A7	34	A8
35	A9	36	A10
37	A11	38	A12
39	A13	40	A14
41	A15	42	
43	* MRDY	44	
45	+5V	46	+5V
47	Q	48	GND
49	E	50	GND

(注)フラット・ケーブル標識線を1とする.

L-8Aバス信号線一覧表

ピン番号	バス信号名(実装面)	ピン番号	バス信号名(半田面)
A	DC+5V	1	DC+5V
B	DC+5V	2	DC+5V
C	DC+5V	3	DC+5V
D	*IRQ	4	G/H(GO/HALT)
E	*NMI	5	*RESET
F	VMA	6	R/W
H	GND	7	Q
J	E	8	GND
K	GND	9	GND
L	RNC	10	RNC
M	DC-12V	11	DC-12V
N	RNC	12	RNC
P	BA	13	REFGRNT
R	*MEMRDY	14	RNC
S	REFCLK	15	DGRNT
T	DC+12V	16	DC+12V
U	RNC	17	RNC
V		18	RNC
W	RNC	19	RNC
X	*MRDYP	20	RNC
Y		21	
Z		22	RNC
a	*FIRQ	23	BS
b	GND	24	GND
c	RNC	25	RNC
d	*EXTMEM	26	
e		27	
f		28	
h	*D3	29	*D1
j	*D7	30	*D5
k	*D2	31	*D0
l	*D6	32	*D4
m	A14	33	A15
n	A13	34	A12
p	A10	35	A11
r	A9	36	A8
s	A6	37	A7
t	A5	38	A4
u	A2	39	A3
v	A1	40	A0
w	GND	41	GND
x	GND	42	GND
y	GND	43	GND

*の付いた信号は負論理
RNCは、システム・リザーブ(使用禁止)

L-8Aバス・インターフェイス

システム拡張ユニットには、システム拡張用のL-8Aバスが用意されています．このL-8Aバスに各種モジュールを実装すれば、システムの拡張ができます．

また、別売のユニバーサル・カード（MB2801）を使えば、自作のモジュールを実装することも可能です

ピン番号	信号名	意味	システム拡張モジュールの入出力
5	＊RESET	●FM-8のシステム・リセット信号です．FM-8本体の電源投入時、またはリセット・スイッチを押している間、LOWになります。	出力
E	＊NMI	●CPUに対する割り込み信号です．この信号が入力されると、CPUは割り込みマスク・ビットに無関係に実行中の命令実行終了後、割り込み処理ルーチンへ分岐します． ●割り込みをかけるには、最低でもCPUの1サイクル分以上の間、LOWである必要があります． (NMI=NON-MASKABLE INTERRUPT)	入力
D	＊IRQ	●CPUに対する割り込み信号ですが、割り込みマスク・ビットによってマスクすることができます ●割り込みをかけるには、最低でもCPUの3サイクル分以上の間、LOWである必要があります． (IRQ=INTERRUPT REQUEST)	入力
a	＊FIRQ	●CPUに対する割り込み信号で、IRQと同様にマスクすることができます．また、優先順位はIRQより上です． ●割り込みをかけるには、IRQと同様に3サイクル分以上LOWである必要があります． (FIRQ=FAST INTERRUPT REQUEST)	入力
d	＊EXTMEM	●拡張ユニット内にメモリを増設したときにFM-8本体内のメモリのアクセスを禁止する信号です．	入力
F	VMA	●定常HIGHレベルの出力信号です	出力
b	R/W	●CPUがメモリ・データのリード・サイクル(H)かライト・サイクル(L)かを示す信号です． (R/W=READ/WRITE)	出力
4	G/H	●この信号をLOWにするとCPUはHALT(一時停止)され、BA、BSは、HIGHレベルになります	入力
J	E	●CPUのクロック信号です．	出力
7	Q	●CPUのクロック信号です．	出力
P 23	BA BS	●BA信号がHIGHのときはCPUのアドレス・バス、データ・ ンピーダンス状態にあることを示します． ●BS信号は、BA信号と併用して下表のようにCPUの状態を示します BA | BS | CPUの 0 | 0 | 通常動作 0 | 1 | 割り込み、またはリセットのAcknowledge 1 | 0 | SYNC命令による待機 1 | 1 | HALTまたはBus Grant	出力
13	REFGRNT	●この信号がHIGHのサイクルはダイナミックRAMにリフレッシュを行なっています．	出力
S	REFCLK	●ダイナミックRAMのリフレッシュのクロック信号です．	出力
R	＊MEMRDY	●低速メモリなどをアクセスするためにCPUクロックEとQをのばすための信号です．	入力
15	DGRNT	●この信号は、DMACからのHALT要求信号に対する応答信号でCPUのバスがハイ・インピーダンス状態となってDMA転送を行なってもよいことを示します． ●この信号がHIGHレベルのときはDMA転送中で、LOWレベルの間は、CPUが動作中です．	出力
X	＊MRDYP	●メモリ・レディをかける必要のあるモジュールで、メモリ・レディ回路をもっていない場合、メモリ・レディ信号にかわるメモリ・レディ要求信号です． ●システム拡張モジュールではこの信号を受け取ると最高2μSECのメモリ・レディ信号をCPUに出力する．	入力
右に示す	＊D0〜＊D7	●CPUとのデータ転送に使うデータ・バスです． ●負論理なので、LOWレベルで"1"になります． ＊D0 ＊D1 ＊D2 ＊D3 31 29 k h ＊D4 ＊D5 ＊D6 ＊D7 32 30 l j	入出力

右に示す	A0〜A15	アドレス・バス A0 A1 A2 A3 A4 ピン 40 v u 39 38 A5 A6 A7 A8 A9 ピン t s 37 36 r A10 A11 A12 A13 A14 ピン p 35 34 n m A15 ピン 33	出力
A、B C、1 2、3	DC+5V	+5V電源ライン	
T 16	DC+12V	+12V電源ライン	
M 11	DC−12V	−12V電源ライン	
b w x y 24 41 42 43	GND	+5V用リターン	
H 8 9	GND	±12V用リターン	

図8　モジュール外形寸法図

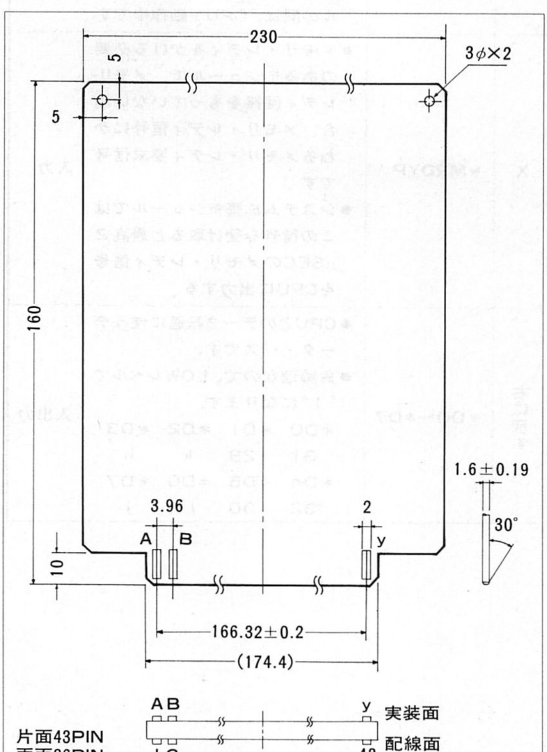
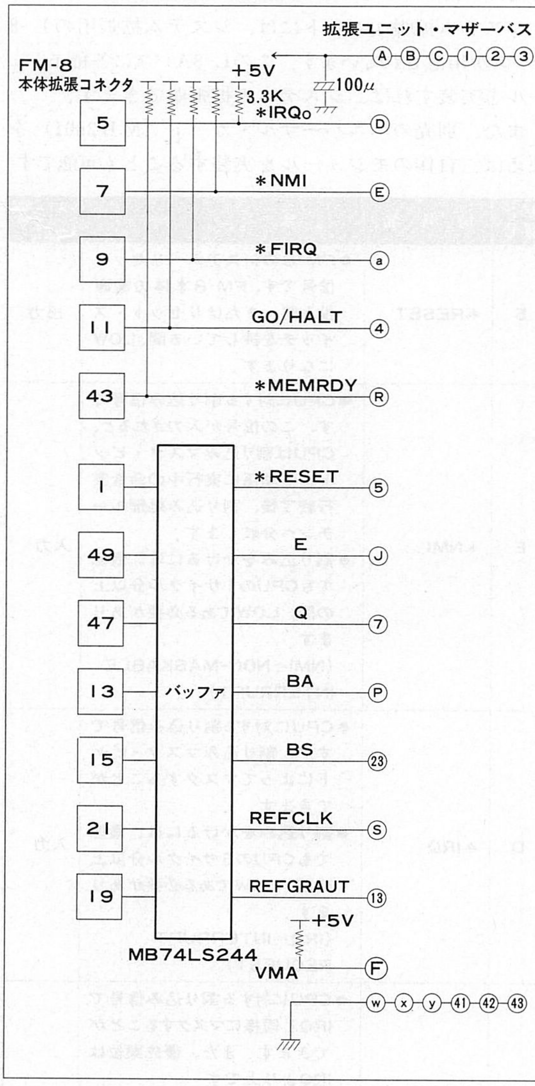


図7　その他信号線の処理方法



BIOS

[1]BIOSの概要

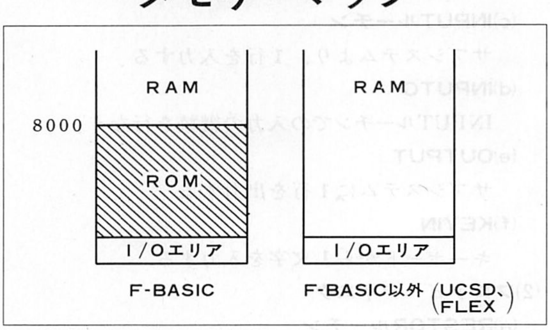
F-BASICシステムでは、I/Oの入出力処理をすべてBIOS(Basic Input Output System)と呼ばれるI/Oドライバ・モジュールを経由して行ないます．ただし、割り込み処理とRS-232Cのポート、ライトペンは扱いません．

また、本BIOSはF-BASIC用で、F-BASIC以外のシステムで利用することはできません．これは、ソフトウェアのインターフェイス上の問題からではなく、BIOSがF-BASICのマスクROM上に存在するということから起こります．

なぜなら、UCSD、PASCAL、FLEXなどはI/Oエリアを除いて、すべてRAM空間で動作するために、BIOSが常駐するROMがないからです．

したがって、本BIOSは通常のI/Oドライバ(ROM常駐)とは性格が異なります．

メモリ・マップ



BIOSがサポートするI/Oデバイスは次のものです．

(1)サブシステム(ディスプレイ＆キーボード)
(2)フロッピーディスク(5.25″、8″)
(3)オーディオ・カセット
(4)プリンタ
(5)ベル(ビープ音)
(6)磁気バブル・カセット
(7)音声合成デバイス
(8)漢字ROM
(9)アナログ入力ポート

これらのI/Oへのアクセスは、BIOSに用意されているドライバ・ルーチンをコールして行ないます．

BIOSのエントリは1箇所だけで、個々のドライバ・ルーチンの識別とか各種パラメータの受け渡しは、RCB(Request Control Block)と呼ばれる8バイトのインターフェイス領域で、すべて行なわれます．このRCBに必要事項をセットして、BIOSをコールするという手順がとられます．

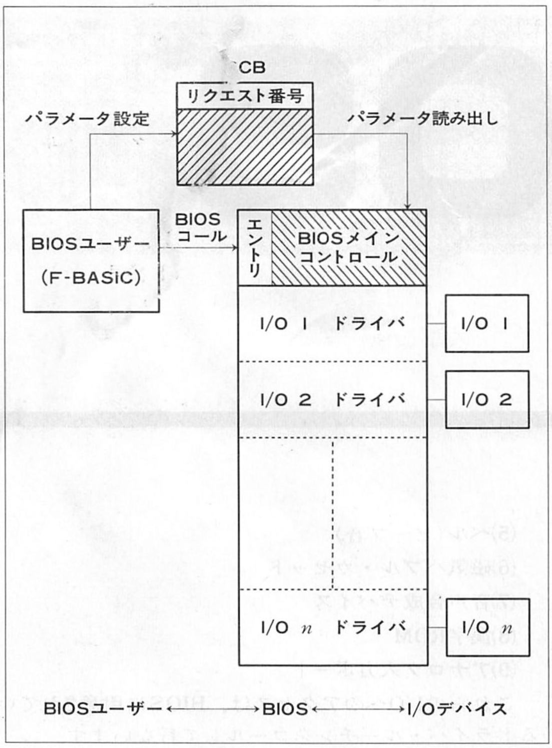
RCBの詳細は後述しますが、RCBの先頭1バイトが、BIOSに対するリクエスト番号を表わしており、この番号により各種のI/Oドライバ・ルーチンのうちから1個が選択され、実行されます．

一般に、I/Oデバイスとのやりとりでは、必ずI/Oエラーの発生があります．このような場合、BIOSは自分自身ではエラーの回復やメッセージの出力はまったく行なわず、単にエラーの内容をユーザーに返すだけです．したがって、ユーザーはBIOSのコール後、RCB中のスティタス・バイトをチェックし、エラーがあれば、それに対応する処理を行なわなければなりません．

このとき、BIOSは各I/Oのエラーを、BIOS独自のエラー番号に変換してユーザーに返します．BIOSのエラー番号については、後に表を示します．フロッピーディスクのリード/ライト・エラーに対するリトライや、プリンタのノット・レディに対する処置も、すべてBIOSのコール側で行なうことが必要です．

BIOSのユーザー、BIOS、I/Oデバイスの関係を図式化すると次のようになります．

BIOSとI/Oの関係図



[2]BIOSの使い方

BIOSの使用手順は次のとおりです．

(1)RCB領域8バイトを設定する．

(2)RCBに、リクエスト番号を始めとして、必要なパラメータを設定する．(設定すべきパラメータは呼び出すルーチンごとに異なる)

(3)インデックス・レジスタXに、RCBの番地をセットする．

(4)BIOSをサブルーチン・コールする．

(5)キャリーフラグにより、エラーがあったかどうかを判定し、エラーがあるときは必要なエラー処理を行なう

なお、BIOSは、PCとCCの1部のフラグを除く、すべてのレジスタの内容を保存します．

CCの中で保存されるのは、E、F、Iの3つです．

BIOSのルーチンでは、必要に応じて、FIRQとIRQのマスクを行なっているので、BIOS実行中はFIRQとIRQの割り込みがかからない場合があります．BIOSの使用手順をコーディングしてみると、次のようになります．

RCBの内容をセットする

```
LDX  #RCB   ………… AccXにRCBのアドレスをセット
JSR  BIOS   ………… BIOSのコール
BCS  ERROR  ………… エラー処理へ分岐
```

BIOSはエラーの有無をキャリーフラグに設定し、エラー番号をRCBのステイタス・バイトに設定する．キャリーフラグは次のように設定します．

C＝0（キャリークリア）……………エラーなし

C＝1（キャリーセット）……………エラーあり

このキャリーフラグは、BIOSを抜けてきた直後に、BCSまたはBCC(LBCS、LBCC) 命令によってチェックすることができます．

RCBは定型の処理に対しては、あらかじめコンスタントでとるほうが効果的です．パラメータがそのつど変化する場合には、むしろ必ず RCB の内容をセットするようにしたほうがよいのです．

RCBの形成はユーザーの自由ですが、諸般の条件を考慮してRCBの内容の設定をできるだけ短いステップに抑えることが望ましいのです．それがBIOSを使うコツです．

[3]BIOSが提供するI/Oドライバ・ルーチン

BIOSに用意されるI/Oドライバ・ルーチンは次のものです．

(1)サブシステム

(a)SUBOUTルーチン

サブシステムに、コマンド・コードやデータを送り出す．

(b)SUBINルーチン

サブシステムに、コマンド・コードやデータを送り出し、サブシステムからの入力を受け取る．

(c)INPUTルーチン

サブシステムより、1行を入力する．

(d)INPUTC

INPUTルーチンでの入力の継続を行なう．

(e)OUTPUT

サブシステムに1行を出力する

(f)KEYIN

キーボードから1文字を入力する．

(2)フロッピーディスク

(a)RESTORルーチン

シーク・トラック0を行なう．

(b)DWRITEルーチン

フロッピーディスクに1セクタを書き込む．

(c)DREADルーチン

フロッピーディスクから1セクタを読み取る．

(3)オーディオ・カセット

(a)MOTORルーチン

カセットのモーターをON、OFFする

(b)CTBWRTルーチン

カセットテープにデータを1バイト書き込む

(c)CTBREDルーチン

カセットテープから1バイトを読み込む．

(4)プリンタ

(a)LPOUTルーチン

プリンタにデータ・ブロックを出力する．

(b)LPCHKルーチン

プリンタのレディ・チェックを行なう．

(c)HDCOPYルーチン

CRT画面をプリンタにコピーする

(d)SCREENルーチン

CRT画面をプリンタにコピーする．

HDCOPYルーチンとの違いは、HDCOPYルーチンは、スクリーン・イメージをできるだけ再現するために、横方向のドットを拡大してプリンタに印字するほか、カラーにより、濃淡を付加しているのに対し、このルーチンは、そういったことはまったく行なっていないという点にあります．

HDCOPYルーチンとSCREENルーチンは、どちらも、一長一短があり、用途による使い分けが肝要です．

(5)ベル

(a)BEEPONルーチン

ベルのスイッチをONする．

(b)BEEPOFルーチン

ベルのスイッチをOFFする。

(6)磁気バブル・カセット

(a)INTBBLルーチン

バブル・カセットをイニシャライズする．

(b)WRTBBLルーチン

バブル・カセットにnページ・ライトする．

(c)REDBBLルーチン

バブル・カセットからnページ・リードする．

(7)音声合成デバイス

(a)VOICEルーチン

音声合成を起動する．

(b)VPARMルーチン

音声合成パラメータをセットする．

(c)VDPUTルーチン

音声合成データを合成部のRAMメモリへ転送する．

(d)VRESETルーチン

音声合成部をリセットする(ソフトウェア・リセット)．

(8)漢字ROM

(a)KANJIRルーチン

漢字ROMから、指定されたJISコードに対応する漢字のドット・パターン32バイトを取り出す．

(9)アナログ入力ポート

(a)ANALGPルーチン

アナログ・ポートからの入力データを読み取る．

(10)その他

BIINIT

BIOSのイニシャライズを行なう．

これらのルーチンには、リクエスト番号と呼ばれる番号が付けられ、そのリクエスト番号により、ルーチンの呼び出しが行なわれる．

BIOSリクエスト番号表

リクエスト番号	ルーチン名	リクエスト番号	ルーチン名
0	ANALGP	17	SUBIN
1	MOTOR	18	INPUT
2	CTBWRT	19	INPUTC
3	CTBRED	20	OUTPUT
4	INTBBL	21	KEYIN
5	SCREEN	22	KANJIR
6	WRTBBL	23	LPCHK
7	REDBBL	24	BIINIT
8	RESTOR	25	VPARM
9	DWRITE	26	VDPUT
10	DREAD	27	VRESET
11	VOICE	28	INPUT
12	BEEPON	29	INPUTC
13	BEEPOF	30	OUTPUT
14	LPOUT	31	KEYIN
15	HDCOPY	32	KANJIR
16	SUBOUT	33	LPCHK

[4]RCBの構成

RCB (Request Control Block) は、BIOSユーザーとBIOSとがコミュニケーションを行なうためのもので、非常に重要な役割を持っています．

RCBは、8バイトで構成されます 先頭の2バイトはすべてのルーチンに共通ですが、残りの6バイトは、各ルーチンごとに使用方法がまちまちで、そのときどきにより持つ意味が異なります

各バイトに与えられる役割のすべての場合について示すとRCBは、次のページの表のように表わせます。

RCBの内容は、BIOSがユーザーに積極的に返すパラメータを除いて保存されます．

RCBのタイプは、次の10通りです

(1)アナログポート・タイプ

(2)カセットモータ・タイプ

(3)カセットリード/ライト・タイプ

(4)ノンパラメータ・タイプ (リクエスト番号とステイタスのみ)

(5)バブルカセット・タイプ

(6)フロッピーディスク・タイプ
(7)プリンタ・タイプ
(8)ハードコピー・タイプ
(9)INPUTルーチン・タイプ
(10)漢字ROMタイプ

以下、10タイプの各パラメータについて、説明します．

RCBの形式

ラベル名	相対値	内容
RQNO	0	リクエスト番号
RCBSTA	1	ステイタス
RCBDBA	2、3	データバッファ・アドレス
RCBCNL	2	アナログポート・チャネル番号
MOTORF	2	カセット・モータON、OFFフラグ
CRWDAT	2	カセットリード/ライト・データ
RCBRNG	3	アナログ・ポート電圧レンジ
RCBCDT	4、5	ハード・コピー色指定データ
RCBAPD	4	アナログ・ポート入力データ
RCBLNH	4	入出力データ長(上位)
RCBTRK	4	フロッピーディスクトラック番号
RCBPAH	4	バブルカセット・ページアドレス(上位)
RCBJCD	4、5	漢字JISコード
RCBLNL	5	入出力データ長(下位)
RCBSCT	5	フロッピーディスク・セクタ番号
RCBPAL	5	バブルカセット・ページアドレス(下位)
RCBSID	6	フロッピーディスクサイド
RCBPCH	6	バブルカセット・ページカウント(上位)
RCBBMH	6	データ・バッファ長(上位)
RCBUNT	7	ディスク・ユニット番号
RCBPCL	7	バブルカセット・ページカウント(下位)
RCBBML	7	データ・バッファ長(下位)

(1)アナログポート・タイプ

ラベル名	相対値	内容
RQNO	0	リクエスト番号
RCBSTA	1	ステイタス
RCBCNL	2	アナログ・ポート・チャンネル番号
RCBRNG	3	電圧レンジ
RCBAPD	4	A/D変換結果

RQNO　：BIOSのサービスルーチン指定番号．
RCBSTA：BIOSのサービスルーチンを実行した結果のステイタスで、BIOSがセットして、ユーザーに返す．
　正常終了時には、バイナリ0にセットされる．それ以外のときはエラー番号がセットされる．

以下、RQNOとRCBSTAの説明は省略します．

RCBCNL：アナログ・ポートのチャンネル番号(0、1、2、3)．
RCBRNG：H($48)…………High Range(0～2.5V)
　L($4C)…………Low Range(0～5/8V)
RCBAPD：A/D変換データ
　符号なし1バイト（0～255)

(適用ルーチン)
ANALGP

(2)カセットモータ・タイプ

ラベル名	相対値	内容
RQNO	0	リクエスト番号
RCBSTA	1	スティタス
MOTORF	2	モータ・フラグ

MOTORF≠FF……………モータをストップする．
　=FF……………モータをスタートする．

(適用ルーチン)
MOTOR

(3)カセット・リード/ライト・タイプ

ラベル名	相対値	内容
RQNO	0	リクエスト番号
RCBSTA	1	スティタス
CRWDAT	2	入出力データ

CRWDAT：リード/ライト・データ1バイト
(適用ルーチン)
CTBWRT、CTBRED

(4)ノンパラメータ・タイプ

ラベル名	相対値	内容
RQNO	0	リクエスト番号
RCBSTA	1	スティタス

(適用ルーチン)
INTBBL、VOICE、BEEPON、BEEPOF、LPCHK、BIINIT、VRESET

(5)バブルカセット・タイプ

ラベル名	相対値	内容
RQNO	0	リクエスト番号
RCBSTA	1	ステイタス
RCBDBA	2、3	データバッファ・アドレス
RCBPAH	4	ページ・アドレス上位
RCBPAL	5	〃　下位
RCBPCH	6	ページ・カウント上位
RCBPCL	7	〃　下位

RCBDBA：リード/ライト・バッファ先頭アドレス
　バッファ長は、(32バイト×ページ数)分必要となる．
RCBPAH、L：バブル・アクセス先頭アドレス
RCBPCH、L：リード/ライト・ページ数

バブル・カセットは容量が32Kバイトであり、1ページが32バイト、$400ページで構成されます．また、バブル・カセットのユニットは2個です．

RCBPAHは、バブル・カセットのページ・アドレスの上位を示すものですが、このバイトの最上位1ビットを使って、バブル・カセットのユニットを表わします．

RCBPAH　　　　RCBPAL

| ・ページ・アドレス上位 |　| ページ・アドレス下位 |

O：バブカセ・ユニットO
I：バブカセ・ユニットI

バブル・カセットのページ・アドレスは$0000から始まります.

RCBPAH、L：ユニット0………$0000～$03FF
　　　　　　ユニット1………$8000～$83FF

バブル・カセットに対して、一度にリード/ライトするページ数をRCBPCH、Lで指定します.

RCBPCH、L：$0001～$0400

ページ・アドレスとページ・カウントの和は$400を越えてはなりません.

(適用ルーチン)

WRTBBL、REDBBL

(6)フロッピーディスク・タイプ

ラベル名	相対値	内容
RQNO	0	リクエスト番号
RCBSTA	1	スティタス
RCBDBA	2、3	バッファ先頭アドレス
RCBTRK	4	トラック番号
RCBSCT	5	セクタ番号
RCBSID	6	サイド番号
RCBUNT	7	ユニット番号

RCBDBA：リード/ライトデータ・バッファアドレス
RCBTRK：トラック番号（0～39）
RCBSCT：セクタ番号（1～16）
RCBSTD：サイド番号（0、1）

7	6	5	4	3	2	1	0
×	×	×	×	×	×	×	

—LSB 1ビットのみ有効

RCBUNT：ユニット番号（0、1、2、3）

7	6	5	4	3	2	1	0
×	×	×	×	×	×		

2ビットのみ有効

(適用ルーチン)

DWRITE、DREAD、RESTOR

(7)プリンタ・タイプ

ラベル名	相対値	内容
RQNO	0	リクエスト番号
RCBSTA	1	ステイタス
RCBDBA	2、3	バッファ先頭アドレス
RCBLNH	4	出力バイト数上位
RCBLNL	5	〃　下位

RCBDBA：出力データ・バッファアドレス
RCBLNH、L：出力バイト数

(適用ルーチン)

LPOUT、SUBOUT、SUBIN、OUTPUT、KEYIN、VPARM、VDPUT

(8)ハードコピー・タイプ

ラベル名	相対値	内容
RQNO	0	リクエスト番号
RCBSTA	1	スティタス
RCBDBA	2、3	バッファ・アドレス
RCBCDT	4、5	色指定データ

RCBCDTは2バイトからなり、コピーするときに、黒でプリントするカラーと灰でプリントするカラーの色指定を行ないます.

RCBCDTが黒でプリントする色の指定を、RCBCDT＋1が灰でプリントする色の指定を行ないます.

RCBCDTの各ビットが、カラーコード（0～7）に対応します.

	7	6	5	4	3	2	1	0	
RCBCDT									……黒プリント指定
＋1									……灰プリント指定
	白	黄	水色	緑	紫	赤	青	黒	

各ビットは、1＝ON、0＝OFFです.

RCBCDTとRCBCDT＋1の両方にビットが立ったときは、そのカラーは黒でプリントされ、両方とも立っていないときは、白でプリントされます（白：プリント無）.

(適用ルーチン)

HDCOPY、SCREEN

(9)INPUTルーチン・タイプ

ラベル名	相対値	内容
RQNO	0	リクエスト番号
RCBSTA	1	スティタス
RCBDBA	2、3	バッファ先頭アドレス
RCBLNH	4	データ長上位
RCBLNL	5	〃　下位
RCBBMH	6	バッファ長上位
RCBBML	7	〃　下位

RCBDBA：入力バッファの先頭アドレス

RCBLNH、L：データ長
ユーザーがセットするのは、出力データ・バイト数で、BIOSがセットするのは入力データのバイト数.

RCBBMH、L：データ・バッファ長
BIOSは、ユーザーのバッファにデータを転送するさいに、バッファ長をチェックし、転送可能かどうかのチェックを行なう.

(適用ルーチン)

INPUT、INPUTC

⑽漢字ROMタイプ

ラベル名	相対値	内容
RQNO	0	リクエスト番号
RCBSTA	1	スティタス
RCBDBA	2、3	バッファ先頭アドレス
RCBJCD	4、5	JISコード

RCBDBA：漢字ROMデータの格納先頭アドレス．
RCBJCD: 漢字JISコード．
(適用ルーチン)
KANJIR

[5]BIOSルーチン使用例

フロッピーディスクのセクタリード・ルーチンとサブシステムからの1文字キー入力ルーチンの使用例を示します．

(1)フロッピーディスクから1セクタをリードする．

リードセクタ・アドレスは、ユニット1、トラック2、セクタ3、サイド1であるとします．

```
RQNO     EQU   0              RCBオフセット
RCBDBA   EQU   2
RCBTRK   EQU   4
RCBSID   EQU   6
*
BIOS     EQU   $F000          このアドレスは一例
                              です.
FDERR    EQU   $E000
*
         LDX   #FDRCB         RCBのアドレスセット
         LDA   #10            リクエスト番号セット
         STA   RQNO,X
         LDD   #FDBUF         バッファのアドレス・セット
         STD   RCBDBA,X
         LDD   #2*256+3       トラック=2、セクタ=3セット
         STD   RCBTRK,X
         LDD   #1*256+1       サイド=1、ユニット=1セット
         STD   RCBSID,X
         JSR   BIOS           BIOSのコール
         LBCS  FDERR          エラーチェック
           :
*
FDRCB    RMB   8              RCB8バイト
*
FDBUF    RMB   256            リード・バッファ256バイト
```

(2)サブシステム・キーボードから1文字入力する．

```
RQNO     EQU   0              RCBオフセット
RCBDBA   EQU   2
RCBLNH   EQU   4
*
KEYBUF   EQU   6              RCBのラスト2バイトを
                              キー入力のためのバッフ
                              ァに使用する.
*
BIOS     EQU   $F000          このアドレスは一例であ
SBERR    EQU   $E003          る.
*
         LDX   #SBRCB         RCBのアドレス・セット
         JSR   BIOS           BIOSのコール
         BCS   SBERR          エラーチェック
           :
*
SBRCB    FCB   21             INKEYリクエスト番号
         FCB   0              ステイタス
         FDB   SBRCB+KEYBUF   キー入力バッファ・アドレス
         FDB   0              RCBLNH、BIOSのリタ
                              ーン時、2がセットされます.
*
*KEY BUFFER
*
         FDB   0              キー入力バッファ
*
```

[6]BIOSルーチンインターフェイス詳細

BIOSの各I/OルーチンのRCBの使い方を以下に述べる．RCBの8バイトのうち、先頭の2バイトはすべてのルーチンについて共通ですが、残りの6バイトは、ルーチンごとに使用方法が異なっています．

リクエスト番号順に、RCBの構成を示します．なお、RCBの構成表中、○印の付いたものが、呼び出し元(ユーザー）あるいはBIOSが設定すべきパラメータです．

RCBの8バイトの内容はBIOSが設定してユーザーに返すパラメータを除いて変化しません．また、BIOSの実行中にエラーが発生した場合、BIOSはそのI/Oルーチンの実行を中止し、エラーステイタスをセットし、呼び出し元に帰ります．そのとき、BIOSがRCBに設定することになっているパラメータの値は正しく設定されているとは限りません．ただし、エラーステイタスだけは正しく設定されています．

(1)ANALGP……アナログ・ポート入力ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	○	
RCBSTA	1		○
RCBCNL	2	○	
RCBRNG	3	○	
RCBAPD	4		○

アナログポートから値を読み取るルーチン．アナログ・データはA/Dコンバータで、デジタル化され、このルーチンにより読み取られます．結果は1バイトの値となります．

ユーザーが指定するパラメータはアナログ・ポートのポート番号とアナログ入力の電圧範囲の２つです．

RCBCNL：０～３　４チャンネルのうちから１つのチャンネルを選択する．

RCBRNG＝H　０V～2.5Vの電圧範囲を指定．

　＝L　０V～5/8Vの電圧範囲を指定．

RCBAPD：A/D変換結果１バイト

ステイタスRCBDBAは常に正常終了を示します．

(2)MOTOR……オーディオカセットモータ・コントロールルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	1	
RCBSTA	1		○
MOTORF	2	○	

オーディオ・カセットのリモート端子により、モータをON、OFFするルーチンです．

モータをONするのか、OFFするのかは、RCB中のMOTORFにセットされる値で決まります．

MOTORH＝FF　モータをONする．

　≠FF　モータをOFFする．

また、モータをON、OFFすることにおいて発生するエラーは考えられないので、ステイタスのRCBSTAは常に、正常終了を示します．

(3)CTBWRT……カセットテープ１バイトライト・ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	2	
RCBSTA	1		○
CRWDAT	2	○	

オーディオ・カセットテープに１バイトのデータをライトするルーチン．ライトすべきデータは、CRWDATに設定します．

このカセットライト・ルーチンと、次のリード・ルーチンは、データのブロッキング、デブロッキングは行なっていません．あくまで、上位のブロック・ライト、ブロック・リードのための１個のサブルーチンです．

これらのルーチンの使用にあたっては、ルーチンの呼び出しに関する時間的制約について充分考慮する必要があります．

(4)CTBRED……カセットテープ１バイトリード・ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	3	
RCBSTA	1		○
CRWDAT	2		○

オーディオカセットテープ１バイトリード・ルーチン

(5)INTBBL……バブルカセットイニシャライズ・ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	4	
RCBSTA	1		○

磁気バブル・カセットをイニシャライズするルーチンです．このルーチンはデバイスのイニシャライズを行なうもので、ディレクトリの初期化というようなたぐいのものとは異なります．どちらかというと、ハードウェアのリセット信号に近い働きをするルーチンです．設定するのはRQNO＝４だけです．

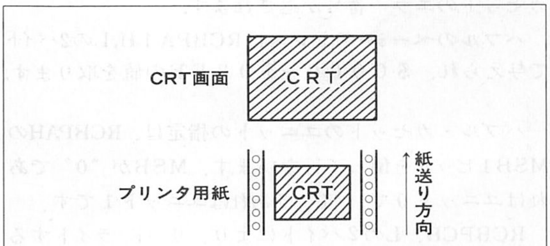
(6)SCREEN……CRTスクリーン・イメージコピー

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	5	
RCBSTA	1		○
RCBDBA	2、3	○	
RCBCDT	4、5		

CRT画面のプリンタへのコピーを行なう．CRT画面上のドットの色とプリンタ上のドットの色は次のように対応します．

CRT画面上	プリンタ上
青、赤、紫、緑、水色、黄、白	→黒
黒	→白

CRT画面とプリンタの紙送り方向との関係は次のとおりです．



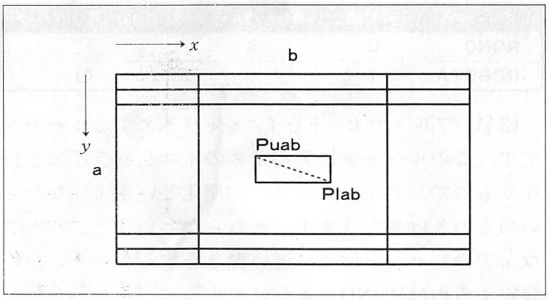
プリンタ上での１画面は、全体的に横広の縮小されたものになります．これは、CRTとプリンタの１ドットのサイズが異なるために起こります．

このSCREENルーチン用のバッファは、RCBDBAにより指定してもらいます．必要なバッファ長は209Dバイトです．

(座標計算)

$$\begin{cases} Puab(x)=80(b-1) \\ Puab(y)=8(a-1) \end{cases}$$

$$\begin{cases} Plab(x)=80b-1 \\ Plab(y)=8a-1 \end{cases}$$



(7)WRTBBL……バブルnページライト・ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	6	
RCBSTA	1		○
RCBDBA	2、3	○	
RCBPAH	4	○	
RCBPAL	5	○	
RCBPCH	6	○	
RCBPCL	7	○	

バブル・カセットにnページをライトします．ライトすべきデータの先頭はRCBDBAで与えられます．

また、バブル・カセット上のライトすべきアドレスはRCBPAH、Lで、ライトすべきページ数は、RCBPCH、Lで与えられます．

RCBSTAには、正常終了ステイタスまたは、バブル・カセットのエラー番号が返されます．

バブルのページ・アドレスはRCBPA1H,Lの2バイトで与えられ、＄0000～＄03FFの値を取ります．

バブル・カセットのユニットの指定は、RCBPAHのMSB1ビットを使って行ないます．MSBが〝0″であればユニット0で、〝1″であればユニット1です．

RCBPCH、Lの2バイトにより、リード/ライトするページ数を与えます．とり得る値は＄0001～＄0400です．

(8)REDBBL……バブルnページ・リード・ルーチン

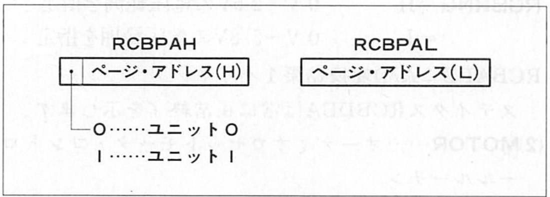
ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	7	
RCBSTA	1		○
RCBDBA	2、3	○	
RCBPAH	4	○	
RCBPAL	5	○	
RCBPCH	6	○	
RCBPCL	7	○	

バブル・カセットからnページをリードします．リード・データの格納先頭アドレスはRCBDBAで与えられます．

また、バブル・カセット上のリードすべきアドレスはRCBPAH、Lで、リードするページ数はRCBPCH、Lで与えられます．

RCBSTAには正常終了ステイタスまたは、バブル・カセットのエラー番号が返されます．

RCBPAHのMSB1ビットは、バブル・カセットのユニットの指定に使われます．



ページ・アドレスは＄0000～＄03FFで、ページ・カウントは＄0001～＄0400です．

(9)RESTORE……シーク・トラック0・ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	8	
RCBSTA	1		○
RCBDBA	2、3		
RCBTRK	4		
RCBSCT	5		
RCBSID	6		
RCBUNT	7	○	

RCBUNTで指定されたディスク・ドライブ・ユニットに対してリストア（シーク・トラック0）動作を行ないます．

これにより、ヘッドはトラック0の位置にセットされます．ドライブ・ユニットの番号は、0、1、2、3です．

(10)DWRITE……フロッピー、1セクタライト・ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	9	
RCBSTA	1		○
RCBDBA	2、3	○	
RCBTRK	4	○	
RCBSCT	5	○	
RCBSID	6	○	
RCBUNT	7	○	

フロッピーディスクに1セクタをライトする．RCBDBAに、ライト・データの先頭アドレスを設定し、RCBTRK、RCBSCT、RCBSID、RCBUNTにより、データを書き込む場所を指定します．各パラメータの指定は、次の範囲で行なえます．

RCBTRK＝0～39（10進）

RCBSCT＝1～16（10進）

RCBSID＝0または1

RCBUNT＝0～3

フロッピーの入出力中にエラーが発生した場合、エラー番号がRCBSTAにセットされる・ユーザーはエラ

ーの内容を見て、リトライするかどうかを決めてやる必要があります

⑾DREAD……フロッピー1セクタ・リード・ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	10	
RCBSTA	1		○
RCBDBA	2、3	○	
RCBTRK	4	○	
RCBSCT	5	○	
RCBSID	6	○	
RCBUNT	7	○	

フロッピーディスクから1セクタをリードします．RCBDBAにリード・データの格納先頭アドレスを設定し、RCBTRK、RCBSCT、RCBSID、RCBUNTにより、リードすべき場所を指定します．各パラメータの指定は、次の範囲で行なえます．

RCBTRK＝0～39（10進）

RCBSCT＝1～16（10進）

RCBSID＝0または1

RCBUNT＝0～3

フロッピーの入出力中にエラーが発生した場合、エラー番号がRCBSTAにセットされます．リトライは、ユーザーの判断に任されます．リトライするときは、リストアを実行し、それからリトライするようにします．

⑿VOICE……音声合成スタート・ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	11	
RCBSTA	1		○

音声合成を開始します．このルーチンに先だって、音声合成のパラメータと音声データが音声合成デバイスに送られていなければなりません．このルーチンは、音声合成をスタートし、音声合成が終了するのを待ってリターンします．

RCBSTAは常に正常終了を示します．

⒀BEEPON……ビープ音発生ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	12	
RCBSTA	1		○

ビープ音の発生を行ないます．RCBSTAは常に正常終了を示します．

⒁BEEPOF……ビープ音ストップ・ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	13	
RCBSTA	1		○

BEEPON で発生中のビープ音をストップします．RCBSTAは常に正常終了をします．

⒂LPOUT……プリンタ出力ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	14	
RCBSTA	1		○
RCBDBA	2、3	○	
RCBLNH	4	○	
RCBLNL	5	○	

プリンタに1データ・ブロックを出力します．このルーチンは、プリンタの行という概念はまったく持っていません．したがって、改行のためのコードもユーザーの側で、印字データとともに送ってやる必要があります．

また、印字中に紙切れ、オフラインなどにより、プリンタがノットレディになった後、再びレディに戻ったとき、行を打ち直すというような処置は行なっていません．

RCBDBAにプリンタに出力するデータの先頭アドレスをセットし、RCBLNH、Lで出力データのバイト数をセットします．

プリンタに1文字出力したいことがありますが、このときは、RCB8バイトのうちの、最後の2バイトをプリント・データバッファに使います．そうすればRCBの他に、バッファを設ける必要はなくなります．

⒃HDCOPY……CRTスクリーンハードコピー・ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	15	
RCBSTA	1		○
RCBDBA	2、3	○	
RCBCDT	4、5	○	

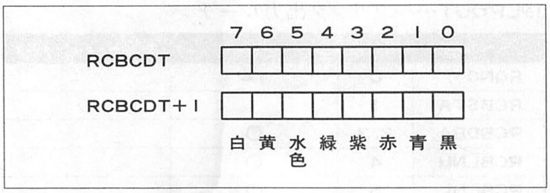
CRTスクリーンをプリンタにコピーする．このコピーでは、スクリーン上の1ドットをプリンタの横方向4ドットに拡大して印字します．この結果、3レベルの濃淡のついたプリントをすることが可能です．スクリーン上では、8色カラーなので、各色に対応した濃淡レベルを実現することは残念ながらできません．

そこで、どの色をどの濃淡に対応させるのかをRCBCDTの2バイトで指定してもらわなければなりません．

RCBCDTは黒レベルでプリントするカラーの指定を、RCBCDT＋1は灰レベルでプリントするカラーの指定を行ないます．

RCBCDTの2バイトで、各ビットがカラーコードを表わし、ビットが立っていれば、その色がその濃淡レベルでプリントされます．

RCBCDT、RCBCDT＋1ともにビットが立っているときには、黒レベルでプリントされます．

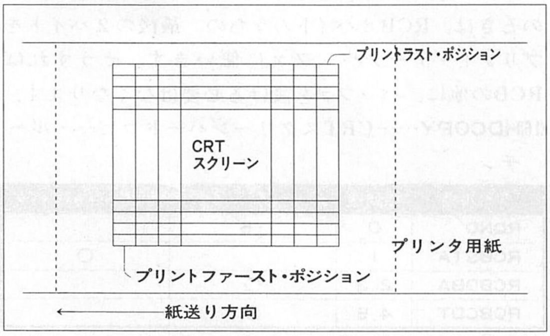


RCBDBAで指定するバッファの長さは210Dバイト以上でなければなりません．

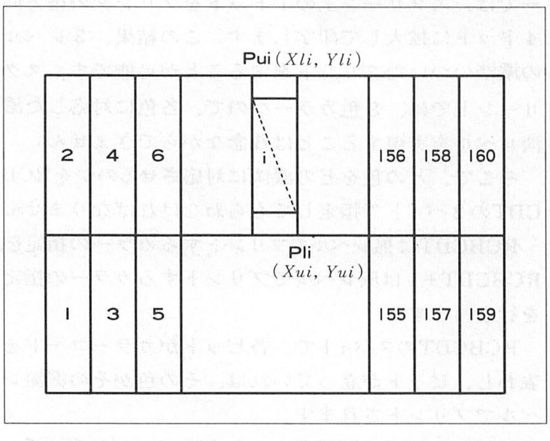
濃淡レベルは黒、灰、白の3つです．これらは、プリンタのドット表現では次のようになります．

レベル	
黒	●●●●
灰	○●○●
白	○○○○

CRT画面とプリント用紙の関係は次のようになります．また、CRTスクリーンのタテ×ヨコの比率とプリンタ用紙上でのタテ×ヨコの比率はほぼ同じです．



(座標計算)



CRT画面からデータを入力するときは、画面を1～160のブロックに分割し、ブロック単位でCRT画面の入力を行ないます．

ブロックの座標（Pui、Pli）の指定は、ブロック番号iにより行なう．Pui、Pliは次のように算出されます．

(1) i＝寄数のとき

Xui＝4 (i－1)　Yui＝100

Xli＝Xui＋7　Yli＝199

(2) i＝偶数のとき

Xui＝4(i－2)　Yui＝0

Xli＝Xui＋7　Yli＝99

⒄SUBOUT……サブシステムアウトプット・ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	16	
RCBSTA	1		○
RCBDBA	2、3	○	
RCBLNH	4	○	
RCBLNL	5	○	

ディスプレイ・サブシステムに、コマンドおよびデータを出力します．一度に出力できるバイト数は最大128バイトです．これは、サブとメインのインターフェイス領域のバイト数です．

このルーチンは、ユーザーがバッファ上に用意したデータをそっくりそのまま、サブに送り出すだけです．したがって、継続フラグの処理などは、ユーザーが行なわなければなりません．

RCBDBAにデータ・バッファの先頭アドレスをセットし、RCBLNH、Lにデータのバイト数をセットする．ただし、RCBLNH＝0とし、RCBLNL＝1～128でなければなりません．

このルーチンは、サブの動作完了を待たずにリターンするので、ステイタスRCBDBAは常に正常終了を示します．

⒅SUBIN……サブシステム・インプット・ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	17	
RCBSTA	1		○
RCBDBA	2、3	○	
RCBLNH	4	○	○
RCBLNL	5	○	○
	6		
RCBMTB	7	○	

サブシステムにコマンドとデータを送り、その結果をデータ・バッファに返す．このルーチンは、サブからの入力データがあるコマンドを実行するときに使います．SUBOUTルーチンを使うのは、サブからの入力データがないコマンドを実行するときです．

このルーチンの場合も、継続フラグの処理などはユーザーが行なわなければなりません．

RCBDBAにデータ・バッファの先頭アドレスをセットし、RCBLNH、Lにデータのバイト数をセットします．このルーチンの実行後は、データ・バッファの内容は入力されたデータになります．サブに出力した、コマンドとデータは失なわれてしまいます．

RCBMTBには、RCBDBAで指定したバッファへ転送バイト数をセットします．BIOSは、指定されたバイト数だけデータを転送します．RCBMTBには0～128の値をセットします．BIOSリターン時には、RCBLNH、Lに、ユーザーバッファへの転送バイト数がセットされます．

⒆INPUT……サブシステム1行入力ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	18	
RCBSTA	1		○
RCBDBA	2、3	○	
RCBLNH	4	○	○
RCBLNL	5	○	○
RCBBMH	6	○	
RCBBML	7	○	

サブシステムのコンソール画面に文字列を表示後、オペレータにより変更されたフィールドのうちで、最上段のフィールドを読み込みます．

サブシステムが変更されたフィールドのデータを転送し始めるのは、終了キーコード(RETURN、CTRL-C、CTRL-X、CLEAR) が入力された直後です．

RCBDBA上にあった画面表示用データは破壊されます．また、RCBLNH、Lの内容もBIOSで新しくセットされます．

入力データはRCBDBAの指すバッファ上に次のように返されます．

←RCBLNH、Lにセットされるバイト数→

| K | 入力されたテキスト |

終了キーコード { 00 RETURN / 01 CTRL-C or CTRL-X / 02 CLEAR

⒇INPUTC……サブシステム継続行入力ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	19	
RCBSTA	1		○
RCBDBA	2、3	○	
RCBLNH	4		○
RCBLNL	5		○
RCBBMH	6	○	
RCBBML	7	○	

INPUTルーチン使用後にコールする．このルーチンは、INPUTで転送されなかった 変更フィールド(未転送フィールド) の転送を行ないます．

INPUTルーチンのコール後は、続けてINPUTCルーチンをコールしなければなりません．INPUTCルーチンは、未転送フィールドがなくなるまでコールし続けなければなりません．

バッファ上での、入力データの返し方はINPUTルーチンの場合と同じです．

←RCBLNH、Lにセットされるバイト数→

| K | 入力されたテキスト |

終了キーコード

(21)OUTPUT……サブシステム・キャラクタ・データ出力ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	20	
RCBSTA	1		○
RCBDBA	2、3	○	
RCBLNH	4	○	
RCBLNL	5	○	

RCBDBAで示されるバッファ上のデータをRCBLNH、Lで指定されたバイト数分だけ、サブシステムのCRT画面に表示します．

(22)KEYIN……キーボード1文字入力ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	21	
RCBSTA	1		○
RCBDBA	2、3	○	
RCBLNH	4		○
RCBLNL	5		○

RCBDBAには、キー入力データ・バッファの先頭アドレスをセットする．バッファは2バイト用意します．

RCBLNH、Lはキー入力データのバイト数を表わし常に＄０００２がBIOSで設定されます．

RCBは8バイトあり、このルーチンの場合後ろの2バイトが空いているので、この2バイトをキー入力データ・バッファとしてもかまいません．

キー入力データ・バッファ

| K | EN |

K：入力キーコード
EN：キー入力の有無　0：無
　　　　　　　　　　1：有

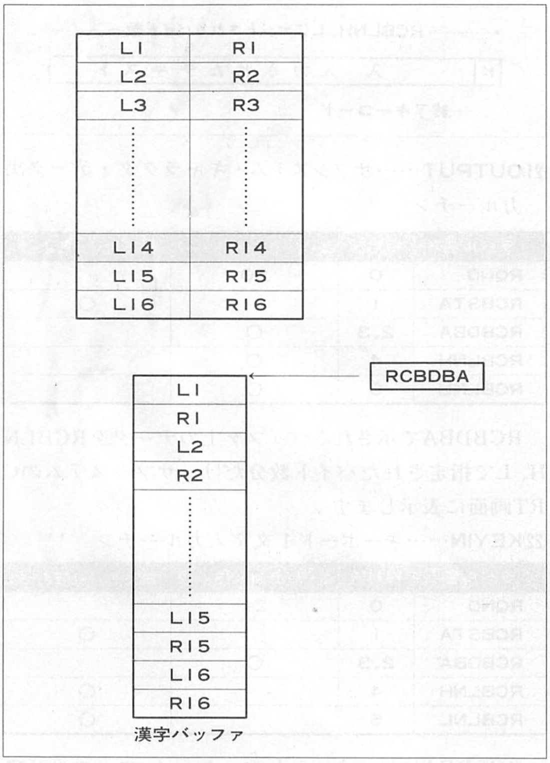
(23)KANJIR……漢字ROMデータリード・ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	22	
RCBSTA	1		○
RCBDBA	2、3	○	
RCBJCD	4、5	○	

漢字データは1字につき32バイトです．空き番号になっているJISコードに対しては、32バイトの0を返す．

漢字1文字のデータ32バイトに対して下図のようにL1～L16、R1～R16なる．ラベルを付けます．

RCBDBAで示されるバッファ上には、L1、R1、L2、R2、……L16、R16の順にデータが格納されます．



(24)LPCHK……プリンタのレディチェック・ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	23	
RCBSTA	1		○

プリンタのレディチェックを行なう．プリンタがレディであれば、正常スティタスを返し、ペーパーエンプティ、ノットレディのときはそれぞれエラー番号をRCBSTAに返します．

RCBSTA＝0　正常（レディ）
50　ペーパーエンプテイ
51　ノットレディ

(25)BIINIT……BIOSイニシャライズ・ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	24	
RCBSTA	1		○

BIOSの初期設定を行なう．このルーチンは、システムの初期設定時にコールします．

(26)VPARM……音声合成パラメータ・セット・ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	25	
RCBSTA	1		○
RCBDBA	2、3	○	
RCBLNH	4	○	
RCBLNL	5	○	

音声合成デバイスに音声合成パラメータを送り出してやるルーチンです．音声合成パラメータの格納領域のアドレスをRCBDBAにセットし，パラメータの長さをRCBLNH、Lにセットする．

音声合成デバイスで発生するエラーはないので、RCBSTAは常に正常終了を示します．

(27)VDPUT…音声合成データ転送ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	26	
RCBSTA	1		○
RCBDBA	2、3	○	
RCBLNH	4	○	
RCBLNL	5	○	

音声合成データを音声合成デバイス部のRAMメモリに転送します．RAMの容量は2Kバイトで、アドレスが＄0000～＄07FFまでです．音声データが2Kバイトを越えて、転送されたときは、転送アドレスが再び＄0000から始まります．つまり，転送アドレスは＄800のモードとなります．

RCBDBAに音声データの先頭をセットし、転送バイト数をRCBLNH、Lにセットします．RCBSTAは常に正常終了を示します．

(28)VRESET……音声合成デバイス・リセット・ルーチン

ラベル名	相対値	ユーザーセット・パラメータ	BIOSセット・パラメータ
RQNO	0	27	
RCBSTA	1		○

音声合成デバイスにリセット信号を送る．

［7］BIOSのエラー番号

BIOSは、各種I/Oのエラーについて、エラー番号を定めており、エラーが起こった場合に、その番号をRCBのステイタス・バイトに返します．

BIOSのエラー番号とI/Oのエラー内容との対応は次のとおりです．

なお、エラーがないときには、0をステイタス・バイトに返します．

(a)0番台……BIOSシステム・エラー
1番　RCBエラー
2番　デバイス・アンアベィラブル・エラー

(b)10番台……フロッピーディスク・エラー
10番　ドライブ・ノットレディ
11番　ディスク・ライトプロテクト
12番　ハード・エラー
（シーク・エラー、ロスト・データ、レコード・ノットファウンド）
13番　CRCエラー
14番　DDマーク検出
15番　タイム・オーバーエラー

(c)30番台……バブルカセット・エラー

30番　未定義コマンド
31番　マーカ未検出
32番　不良ループ数が多い
33番　データ・トランスファ・エラー
34番　CRCエラー
35番　ページ・アドレシング・エラー
36番　(空き)
37番　デバイス・ノットレディ(バブル・カセット未装着)

(d)50番台……プリンタ・エラー、オーディオカセット・エラー

50番　プリンタ・ペーパーエンプティ
51番　プリンタ・ノットレディ
52番　オーディオカセット・リードエラー

(e)60、70番台……サブシステム・エラー

60番　INITコマンドのパラメータに誤りがある
61番　コンソール座標エラー
62番　複数バイトのオーダSEQで必要なデータがない
63番　グラフィック座標エラー
64番　使用できないファンクション・コード、または未定義のファンクション・コード
65番　座標数が規定値より多い、または少ない
66番　文字数が規定値より多い、または少ない
67番　色指定数が規定値より多い、または少ない
68番　ファンクション・キー番号エラー
69番　パラメータ・エラー
70番　コマンド・エラー

[8] BIOSルーチン使用上の留意点

(1)BIOSのワーク・エリア

BIOSのワーク・エリアは＄ＦＦＥ０～＄ＦＦＥＦの16バイトに設定されています.

この16バイトは、フロッピーのアクセスとサブ入出力のときに使われます.

BIOSを使うときは、これらのRAM領域をユーザーが使ってはなりません.

(2)ハードコピー・ルーチンのバッファの確保

HDCOPY、SCREENルーチンは209バイトのワーク領域が必要です. 209バイト以上のエリアを確保し、そのアドレスをRCBDBAにセットしてください.

(3)FIRQ、IRQのマスク

BIOSは必要に応じてFIRQとIRQをマスクします.したがって、BIOS実行中はFIRQ、IRQがかからないことがあります.

(4)カセットのリード/ライト

カセットの1バイトリード/ライト・ルーチンの使用にあたっては、時間的な制約を充分に考慮しなければなりません.

(5)フロッピーディスク・リード/ライト

フロッピーディスクのリード/ライト・ルーチンは、BASICのマスクROMではなく、ブート用のROM（MB8516）上にのせます. これは、ディスクが8インチになったときに対応することと、BIOSが使用するマスクROMのエリアをできるかぎり少なくするためです.

ディスクについては、5″と8″はBASICで同時に扱うことはないという前提です.

また、5″と8″では、ブート用のROMが別物になるという前提で設計しています.

(6)バブル

バブル・カセットは、32Kバイトのみを考慮しており、128Kバイトのバブル・カセットの場合は、プログラムの変更が必要となります.

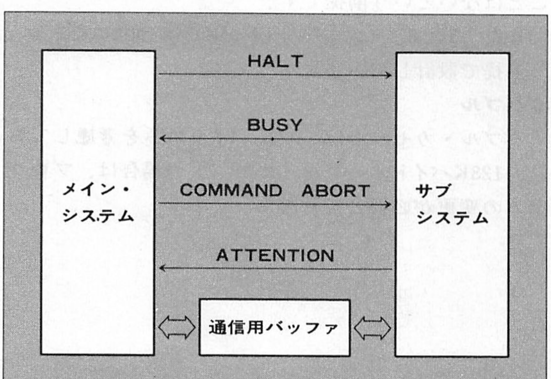
ディスプレイ・サブシステム

第1章　概説

FPC-Xでは、中央処理装置の負荷を軽減するためにDisplay、Keyboardの制御を専用のプロセッサで行ないます．この部分をDisplay Sub-systemと呼びます．

§1.インターフェイス

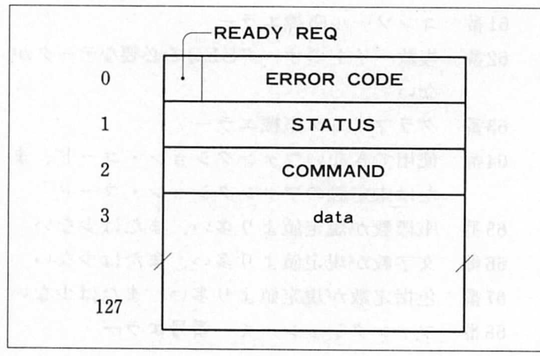
サブシステムとメイン・システムの情報交換は、通信用バッファ(Communication Buffer)と制御線で行なわれます．



1 通信用バッファ

(Communication Buffer)

コマンド・データの交換に使われます．



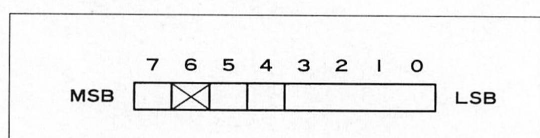
a. READY REQ

SUBSYSTEMをHALTし新たにコマンドを設定しない場合、このbitをONにし､BUSY Lineを落す (Ready).

b. ERROR CODE

コマンド・コードおよびデータに誤りがあった場合に設定される．正しく処理された場合は0．

c. STATUS



0～3	PF割り込みコード　1～10	割り込み処理参照	
4	TIMER割り込み	割り込み処理参照	
5	INTERVAL TIMER割り込み	割り込み処理参照	
6	0時 割り込み	割り込み処理参照	
7	データ継続フラグ	各コマンド参照	

d. COMMAND　　各コマンド参照

e. data　　各コマンド参照

コマンド一覧表

機能ブロック	コマンド名	コード16進
コンソール	INIT	01
	ERASE	02
	PUT	03
	GET	04
	GETC	05
	GET Character Block 1	06
	PUT Character Block 1	07
	GET Character Block 2	08
	PUT Charater Block 2	09
	GET Buffer Address	0A
	TAB SET	0B
	CONSOLE CONTROL	0C
	ERASE 2	0D
	CHARACTER LINE	20
グラフィック	LINE	15
	CHAIN	16
	POINT	17
	PAINT	18
	SYMBOL	19
	CHANGE COLOR	1A
	GET BLOCK 1	1B
	PUT BLOCK 1	1C
	GET BLOCK 2	1D
	PUT BLOCK 2	1E
	GRAPHIC CURSOR	1F
キーボード	INKEY	29
	Define string of PF	2A
	GET string of PF	2B
	Interrupt Control	2C
タイマー	SET TIMER	3D
	READ TIMER	3E
メンテナンス	TEST	3F
	CONTINUE	64

エラーコード一覧表

エラーコード 10進 (16進)	内容
60(3C)	INITコマンドのパラメータに誤りがある.
61(3D)	コンソール座標値に誤りがある.
62(3E)	オーダ・シーケンスに誤りがある.
63(3F)	グラフィック座標値に誤りがある.
64(40)	ファンクション・コードに誤りがある.
65(41)	座標数に誤りがある.
66(42)	文字数に誤りがある.
67(43)	色数に誤りがある.
68(44)	PFキー番号に誤りがある.
69(45)	コマンド・パラメータに誤りがある.
70(46)	コマンド・コードに誤りがある

2 HALT

サブシステムの動作を停止させ、メイン・システムから通信用バッファへの参照を可能にします.

3 BUSY

サブシステムがコマンド実行中であることを通知します.

4 COMMAND ABORT

コマンド動作を中止させる割り込み要求です.(1・2 割り込み処理参照)

5 ATTENTION

メインから依頼のあった割り込み事象が発生したことを通知します(1・2割り込み処理参照).

§2.割り込み処理

サブシステムはメイン・システムに対して次の割り込み機能を持っています.

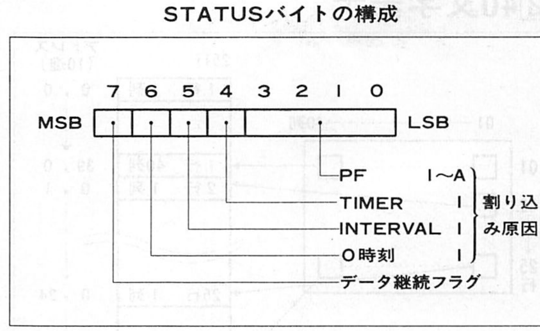
1 COMMAND ABORT (メイン→サブ)

GETおよびGRAPHIC CURSORコマンド実行中の場合に、コマンドの実行を中止し、ただちにコマンド待ちになります.

2 ATTENTION (サブ→メイン)

サブシステムは、割り込み原因を通信用バッファ中のSTATUSバイトに設定し、メイン・システムに割り込みをかけます. サブシステムではSTATUSバイト中の各原因のパートに0でない数を書き込むだけなので、メインの割り込み処理ルーチンは、割り込みを受け付けたとき、または必要でなくなったときに割り込み原因の各パートを0にする必要があります (注:データ継続フラグを変更してはならない).

STATUSバイトの構成



第2章 コンソール

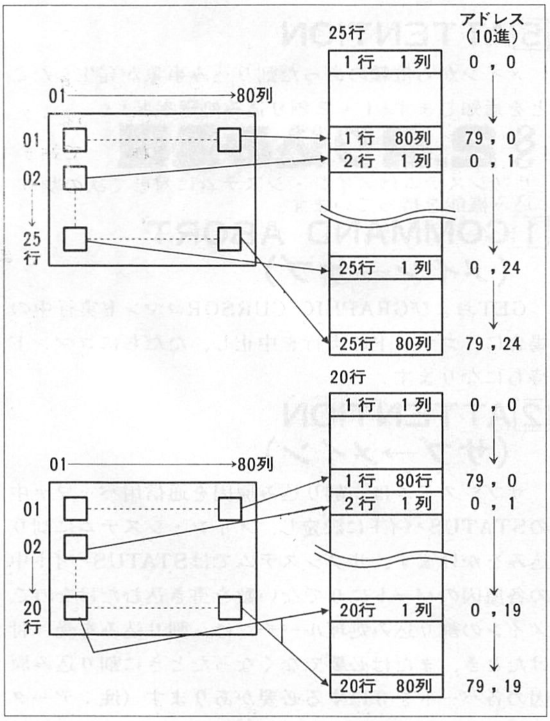
§1.表示

1.表示動作

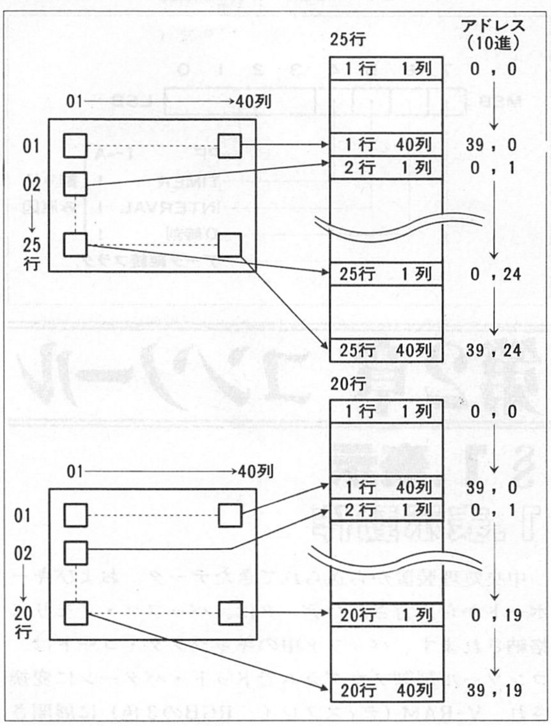
中央処理装置から送られてきたデータ、およびキーボードから入力されたデータは、バッファ・メモリに格納されます. バッファ中のキャラクタ・コードは、コンソール制御プログラムでドット・パターンに変換され、V-RAM (ディスプレイ、RGBの3色) に展開されます.

バッファ・メモリのアドレスとCRT上の文字表示位置とは2種類の対応があります (アドレスは列、行の2次元で表わす).

1 80文字表示

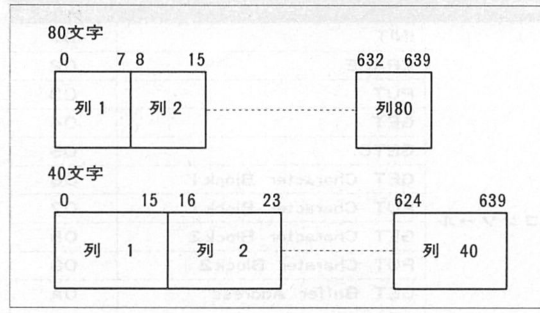


2 40文字表示

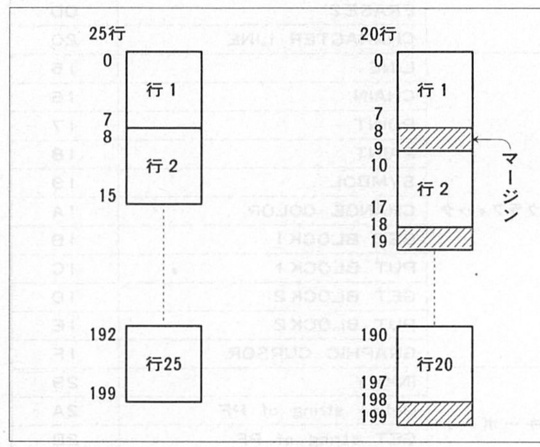


コンソール動作での文字位置とグラフィック画面の座標位置の関係を示す．

横軸



縦軸

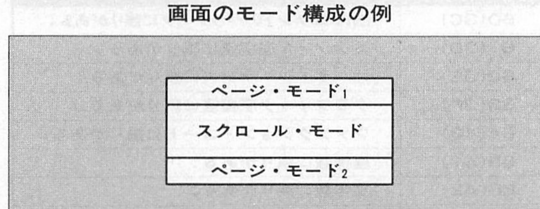


2. コンソール動作の画面制御

1 動作モード

Display Sub-systemのコンソール動作には、画面制御にスクロールとページがあり、1画面中に混在してもよい（ただし、スクロール範囲は1つだけ）.

画面のモード構成の例



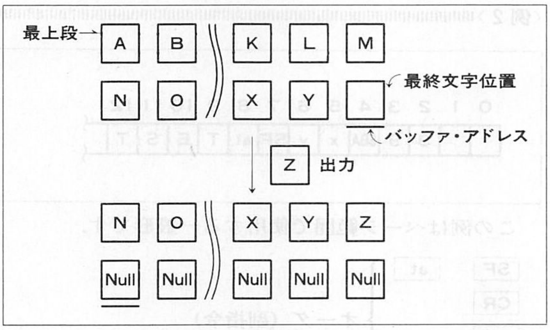
オペレータで、カーソルが移動できる範囲は、そのときのバッファ・アドレスを含む画面モードの範囲です．

(注)PF定義文字列表示を指示する場合、最終行とその手前の2行は、スクロール、ページのいずれにも使用できません．

バッファ・アドレスとは、コマンドで文字を表示する位置です．バッファ・アドレスはコマンド(画面の消去)やオーダ(SBA、EA、NL…)などで設定され、文字を表示するたびに1加算されます．オペレータによるカーソルの移動では次の場合に限り、バッファ・アドレスは変化します．オペレータが挿入した文字位置(Y＊n＋X、nは横文字数)がバッファ・アドレスより大きいか等しい場合にバッファ・アドレスは変化し、その文字の次を示します．

(a)スクロール・モード画面

スクロール・モード画面では、その画面範囲の最終文字に文字を出力(コマンドおよびオペレータ)した場合、画面範囲の第2行から最終行までの行は1行上へ移動し、最下段はNullが挿入されます．このことで最上段であった行は画面から消えることになります．また、バッファ・アドレスは最下段先頭文字位置に移動します．



(b)ページ・モード画面

ページ・モード画面では、その画面範囲（最大2つ、独立）の最終文字位置にコマンドにより文字を表示した場合、バッファ・アドレスは先頭文字に移動します．



オペレータにより最終文字位置に文字が挿入された場合は、画面範囲の先頭から最初の非保護フィールドの先頭にバッファ・アドレスは移動します．

2 フィールド

フィールドとは、一連の文字列を単位としてプログラムで定義され、データの論理処理、表示を行なう最小の単位です．

フィールドの先頭はアトリビュート・バッファ中のアトリビュート文字(Attribute character、次項参照)で、フィールド・スタート・ビットがONであるアトリビュート文字に対応する文字（表示される）で始まり、その終りは次のフィールドの先頭の直前かそのフィールドの属するモード画面の終りまでです．

プログラムでフィールドが定義されていないか、またはスクロールでフィールドの先頭を失なった残りの部分（モード別の画面範囲の先頭にフィールドの先頭が無い）は、非保護フィールドとして扱います．

3 アトリビュート文字

コンソールのバッファには文字コード・バッファとアトリビュート・バッファがあります．文字コード・バッファには文字コード（JIS 8 単位）があり、アトリビュート・バッファには該当する文字の性格を表わす情報が書き込まれます．この情報をアトリビュート文字と呼びます．

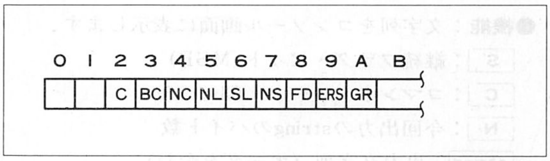
アトリビュート文字の構成を示します．



アトリビュート文字はコンソール出力文字列中のオーダー（副指令）としてSubに送られ、アトリビュート・バッファに設定されます．

§2.コンソール・コマンド

1 INIT



●機能：コンソール機能の初期化をします

C	：コマンド・コード01_{16}	(1)
BC	：背景色	(0～7)
NC	：桁数	(80または40)
NL	：行数	(25または20)
SL	：スクロール開始行番号	

NS：スクロール行数

FD：ファンクション・キー表示　0＝非表示　≠0　表示

ERS：初期設定後の画面の消去　0＝しない　≠0　する

GR：単色表示(グリーン)　0＝しない　≠0　する

(注)　FD＝1の場合、最終2行はPF表示用として確保されるので、SLおよびNSによるスクロール・ウィンドウの指示はこの部分を含んではいけません．

●復帰情報：エラーコード（エラーの場合、RESET時のパラメータで初期化）．

RESET時のパラメータ

BC＝0、NC＝80、NL＝25、SL＝0、NS＝25、FD＝0、ERS＝1、GR＝0

2 ERASE

0	1	2	3	4
		C	W	BC

●機能：画面およびコンソール・バッファを初期化します．

C：コマンド・コード　02_{16}　(2)

W：範囲　0　全画面
　　　　1　スクロール
　　　　2　ページ1
　　　　3　ページ2

BC：背景色　0～7 (W＝0 有効)

●復帰情報：エラーコード

3 PUT

0	1	2	3	4	5
	S'	C	N	string	

●機能：文字列をコンソール画面に表示します．

S'：継続フラグ・ビット(MSB)

C：コマンド・コード　03_{16}　(3)

N：今回出力のstringのバイト数

string：出力文字列（オーダを含む）

●復帰情報

0	1	2
E		0

E：エラーコード（オーダ・エラー）

出力文字列が124バイトを越える場合や、処理上分割したい場合は、S＝'1'でコマンドを発信します．残りの出力文字列は次の形式で転送します．

0	1	2	3	4
	S'	100	N	string

S'＝'0'まで連続した文字列として扱かわれます．

●コマンドの使用例

〈例1〉

0	1	2	3	4	5	6	7	8	9	10	11
		3	8	SF	at	T	E	S	T	CR	LF

この例はスクロール範囲で使用する一般形です．

〈例2〉

0	1	2	3	4	5	6	7	8	9	10	11	12
		3	9	SBA	x	y	SF	at	T	E	S	T

この例はページ範囲で使用する一般形です．

SF　at
CR
LF
SBA　*x*　*y*
　} オーダ（副指令）

4 GET

0	1	2	3	4
	S'	C	N	string

●機能：文字列をコンソール画面に表示後、オペレータから変更されたフィールドのうちで、最上段のフィールドを読み込みます．カーソルの表示位置は、文字列・表示後のバッファ・アドレスの位置です．

S'：継続フラグ・ビット（MSB）

C：コマンド・コード　04_{16}　(4)

N：今回出力のstringのバイト数

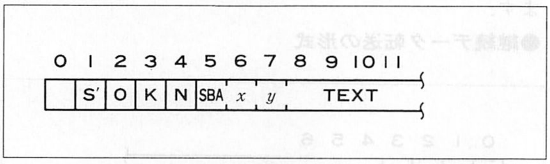
string：出力文字列(オーダを含む)

出力文字列の継続法は3PUTコマンドと同じです．

S'＝'0'の文字列の表示が終ると、オペレータによ

るKey入力が可能になります．

●復帰情報



S′：継続フラグ・ビット（MSB）

K：終了キーコード（00：RETURN、01：CTL-C or, CTL-X、02：CLEAR）

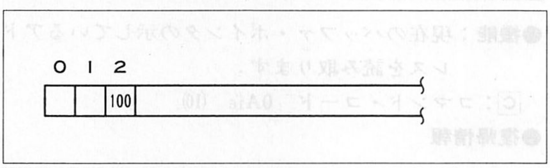
N：今回転送分の変更フィールドデータ・バイト数

SBA x y：変更フィールドの先頭座標

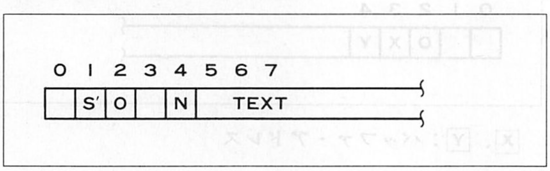
TEXT：フィールド中の制御コード（00～1F、7F）を除く文字列

（注）変更フィールドがない場合、N＝0
　変更フィールドが空文字列の場合、N＝3
　（SBA x yのみ）

変更フィールドの文字列長が長い場合、S′＝"1"として分割転送を行なうので、メイン・システムは次のように残りを要求します．



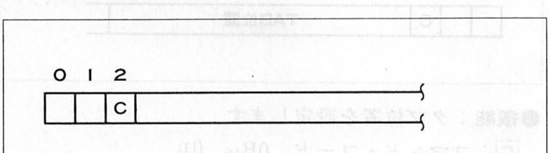
●復帰情報



以下、S′＝"0"まで繰り返します．

SBA x yがないことに注意してください．

5 GETC

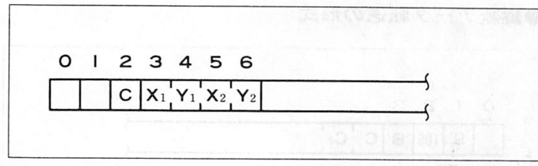


●機能：GET後オペレータによる修正されたフィールドで未転送のフィールドを転送します．

C：コマンド・コード　05_{16}　(5)

●復帰情報：4 GETと同じ．

6 GET CHARACTER BLOCK1

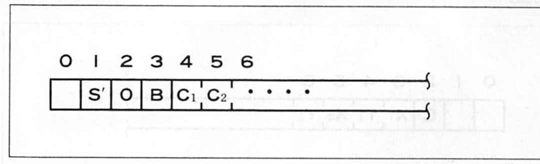


●機能：対角線座標で示される枠内の文字コードを読み取ります．

C：コマンド コード　06_{16}　(6)

X_1 Y_1 － X_2 Y_2：対角線座標（コンソール）

●復帰情報

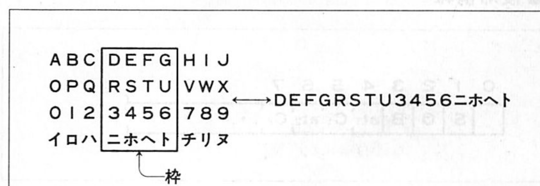


S′：継続フラグ・ビット（MSB）

B：今回転送バイト数

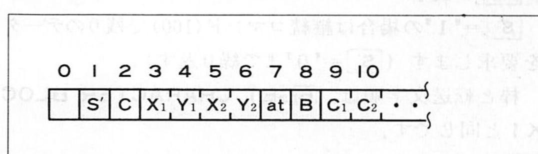
Cn：文字

S′＝"1"の場合は継続コマンド（100）で残りのデータを要求すること（S′＝"0"まで繰り返す）．



枠と転送文字順

7 PUT CHARACTER BLOCK 1



●機能：文角線座標で示される枠内に文字を表示します．

S′：継続フラグ・ビット（MSB）

C：コマンド・コード　07_{16}　(7)

X_1 Y_1 － X_2 Y_2：対角線座標（コンソール）

at：アトリビュート（色指定）

B：今回転送バイト数

Cn：文字

分割転送の場合はS′＝"1"にてコマンドを発信すること．

●継続データ転送の形式

0	1	2	3		
	S′	100	B	C_1	C_2

以下 S′ ＝'0'まで繰り返します．

枠と文字列順は、6GET CHARACTER BLOCK 1と同じです．

8 GET CHARACTER BLOCK2

0	1	2	3	4	5	6
		C	X_1	Y_1	X_2	Y_2

●機能：対角線座標で示される枠内のアトリビュートと文字を読み取ります．

C ：コマンド・コード　08_{16}　(8)

X_1 Y_1 − X_2 Y_2 ：対角線座標(コンソール)

●復帰情報

0	1	2	3	4	5	6	7	
	S′	O	B	at_1	C_1	at_2	C_2	・・・

S′ ：継続フラグ・ビット（MSB）

B ：今回転送バイト数

at*n* ：アトリビュート

C*n* ：文字

S′ ＝'1'の場合は継続コマンド(100)で残りのデータを要求します（S′ ＝'0'まで繰り返す）．

枠と転送文字順は、6GET CHARACTER BLOCK1と同じです．

9 PUT CHARACTER BLOCK2

0	1	2	3	4	5	6	7	8	9	10	11	12	
	S′	C	X_1	Y	X_2	Y_2	7キ	B	at_1	C_1	at_2	C_2	・・

●機能：対角線座標で示される枠内にアトリビュート付の文字を表示します．

S′ ：継続フラグ・ビット（MSB）

C ：コマンド・コード　09_{16}　(9)

X_1 Y_1 − X_2 Y_2 ：対角線座標(コンソール)

B ：今回転送バイト数

at*n* ：アトリビュート

C*n* ：文字

分割転送の場合は、S′ ＝'1'にてコマンドを発信します．

●継続データ転送の形式

0	1	2	3	4	5	6
	S′	100	B	継続データ		

(注)継続データは前転送が at 、 C のいずれで終了したかにより、 at 、 C のいずれかで始まる．

以下、S′ ＝'0'まで繰り返す．

枠と文字順は、6GET CHARACTER BLOCK 1と同じです．

10 GET Buffen Address

0	1	2
		C

●機能：現在のバッファ・ポインタの示しているアドレスを読み取ります．

C ：コマンド・コード　$0A_{16}$　(10)

●復帰情報

0	1	2	3	4
		O	X	Y

X 、 Y ：バッファ・アドレス

11 TAB SET

0	1	2	3	4	5	6	7	8	9	10	11	12
		C	TAB位置									

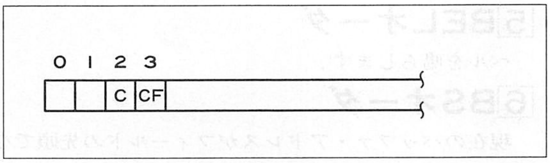
●機能：タブ位置を設定します．

C ：コマンド・コード　$0B_{16}$　(11)

TAB位置 ：10バイト（80bit）を画面の１行分のイメージとし、ONの位置をTAB停止位置とする．先頭バイトのMSBをコンソール座標Xの0、最終バイトのLSBをコンソール座標Xの79と考えます．

●復帰情報：なし

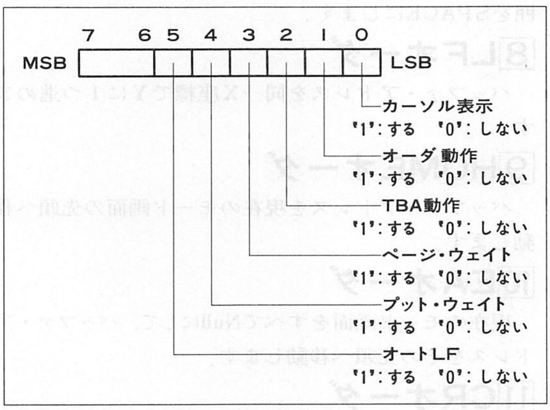
12 CONSOLE CONTROL



●**機能**：種々のコンソール機能を選択する．

C ：コマンド・コード　0C16　(12)

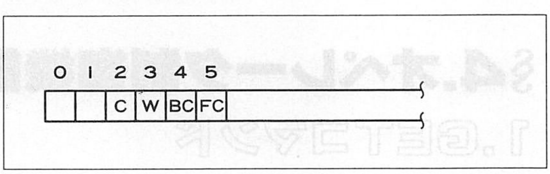
CF ：制御フラグ



RESET時　23_{16}

●**復帰情報**：なし

13 ERASE 2



●**機能**：画面およびコンソール・バッファを初期化します．

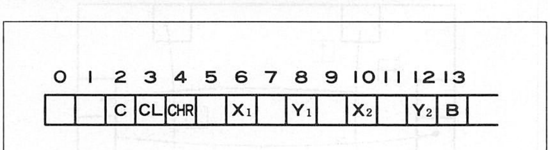
C ：コマンド・コード　$0D_{16}$　(13)

W ：範囲　0 全画面
1 スクロール
2 ページ1
3 ページ2

BC ：背景色0～7（W＝0のときのみ有効）

FC ：文字色0～15

14 CHARACTER LINE



●**機種**：X1・Y1 から X2・Y2 まで文字で直線を引くか、または四角を描きます．

C ：コマンド・コード

CL ：カラーコード

CHR ：文字コード

X1 、Y1 ：コンソール座標1

X2 、Y2 ：コンソール座標2

B ：ボックス・フラグ（0：直線　1：枠　2：四角形塗り）

●**復帰情報**：なし

(注)座標はコンソールですが、コンソール・バッファは変化しません．

§3.オーダ

オーダとは、コンソール出力のデータ列中で書き込みアドレスを指定したり、フィールドを定義したりするための副指令です．

オーダは、00_{16}～$1F_{16}$、$7F_{16}$の範囲のコードですが、次項以下に記述のないものは何も動作しません（No operation Order）．また、コンソール動作制御でオーダ動作をしないよう指示されている場合は、すべての値（00_{16}～FF_{16}）がデータとして扱われて表示されます．

1.オーダの種類

オーダ ＼ シーケンス	1 (オーダ・コード)	2	3
Start Field (SF)	X'11'	アトリビュート	
Set Buffer Address (SBA)	X'12'	X座標	Y座標
Repeat Character (RC)	X'13'	文字数	文字コード
Erose Line (EL)	X'05'		
Bell (BEL)	X'07'		
Back Space (BS)	X'08'		
Horizontal Tab (HT)	X'09'		
Line Feed (LF)	X'0A'		
HOME	X'0B'		
Erase All (EA)	X'0C'		
Carriage Return (CR)	X'0D'		

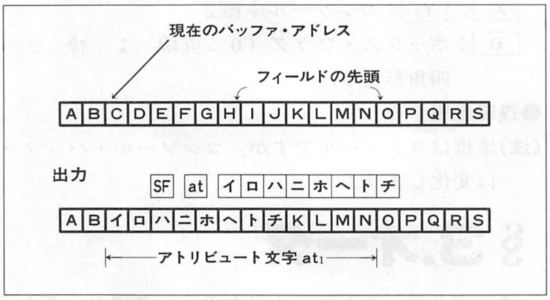
2.オーダの機能と動作

1 SFオーダ

SFオーダは2バイトからなるオーダ・シーケンスで、バッファ中にフィールドを定義するために使います．SF文字に続く1バイトはアトリビュート文字としてアトリビュート・バッファの現在のバッファ・アドレスから、このとき形成されるフィールドの終りまでの間

のすべてに設定されます．このオーダ自身でバッファ・アドレスは変化しません．

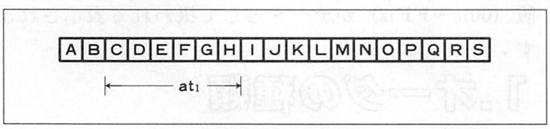
〈例〉



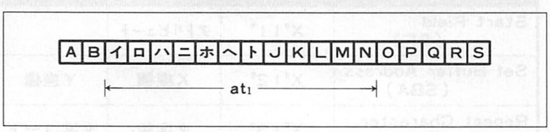
この例では出力文字列によってフィールド範囲を変えています．イ～Nまでat₁のアトリビュートとなりますが、KLMNの表示色は以前のままです．

at₁の設定は次の段階を経ています．

①SFオーダによる



②一連の文字列が①の範囲を越えた



※『一連の文字列』とは、SFオーダに続く文字データ、BELオーダ、BSオーダ、HTオーダ、RCオーダの並びのことです．

2 SBAオーダ

SBAオーダは、3バイトからなるオーダ・シーケンスで、一般データの書き込み、特定のオーダの動作に先立ってバッファ・アドレスを確定するために使います．

バッファ・アドレスは、横軸（X、0～79）*、縦軸（Y、0～24）**で構成されます．

*　X値の上限は、39、79の2種類

**　Y値の上限は、19、24の2種類

3 RCオーダ

3バイトからなるオーダ・シーケンスで、現在のバッファ・アドレスより指定個数分指定文字を表示します．

4 ELオーダ

現在のバッファ・アドレスからそのフィールドの終りまでをNullで埋めます．バッファ・アドレスは変化しません．

5 BELオーダ

ベルを鳴らします．

6 BSオーダ

現在のバッファ・アドレスがフィールドの先頭でないならバッファ・アドレスを1つ前に戻します．

7 HTオーダ

コンソール動作制御がタブONならば、次のタブ停止位置までバッファ・アドレスを移動させ、移動した範囲をSPACEにします．

8 LFオーダ

バッファ・アドレスを同一X座標でYに1つ進めます．

9 HOMEオーダ

バッファ・アドレスを現在のモード画面の先頭へ移動します．

10 EAオーダ

現在のモード画面をすべてNullにして、バッファ・アドレスをその先頭へ移動します．

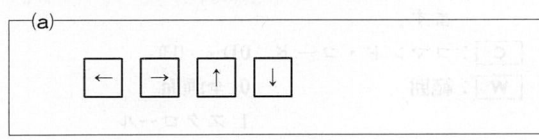
11 CRオーダ

バッファ・アドレスを現在の行の先頭へ移動します。また、コンソール制御でオートLFが選択されている場合は、8 LFオーダの動作も行ないます．

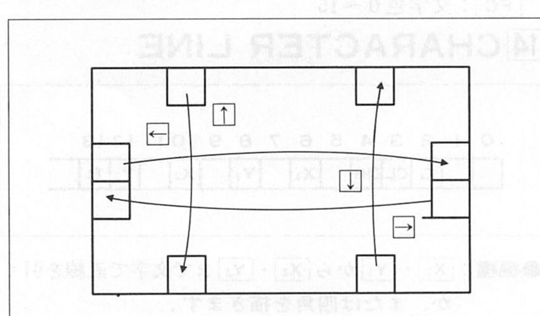
§4.オペレータ制御機能

1.GETコマンド

1 カーソル移動



これらのキーは、カーソルを行または、文字単位で指定された方向へ移動させます．いずれのキーも同一行または同一桁でラップします．



コントロール・シフトによる同一機能キー

← ＝ CTRL ─]

→ ＝ CTRL ─ ￥

↑ ＝ CTRL ─ ^

↓ ＝ CTRL ─ −

(b)

SHIFT ─ ←　　SHIFT ─ →

これらのキーはカーソルを英数文字列の先頭へ移動させるものです．これらのキーは画面の先頭と終端でラップします．

〈例〉

カーソル

[F]UJITSU LTD. TEL 777−1111

↓ SHIFT ─ →

FUJITSU [L]TD. TEL 777−1111

↓ SHIFT ─ →, SHIFT ─ →

FUJITSU LTD. TEL [7]77−1111

↓ SHIFT ─ ←

FUJITSU LTD. [T]EL 777−1111

▶コントロール・シフトによる同一機能キー

SHIFT ─ ← ＝ CTRL ─ B

SHIFT ─ → ＝ CTRL ─ F

(c)

SHIFT ─ ↑　　SHIFT ─ ↓

これらのキーはカーソルをフィールドの先頭に移動させるもので、SHIFT─↑は、前方へSHIFT─↓は、後方へそれぞれ移動します．これらのキーは画面の先頭と終端でラップします．

▶コントロール・シフトによる同一機能キー

SHIFT ─ ↑ ＝ CTRL ─ Y

SHIFT ─ ↓ ＝ CTRL ─ Z

(d)

HOME

このキーはカーソルを画面の先頭に移動するものです．

2 編集キー

(a)

INS　(INSERT MODE)

このキーの押下により、キーボードは挿入モードに設定され、キーボード上のINSERT MODEインジケータが点灯（表示ONのとき）し、文字の挿入動作が可能になります．

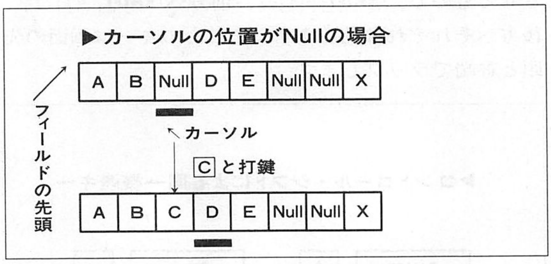
挿入動作はカーソル位置によって以下のように行なわれます．

①カーソルの示すアドレスの内容がNullである場合、データ・キーを打鍵するとカーソルの位置にその文字ががが入ります．

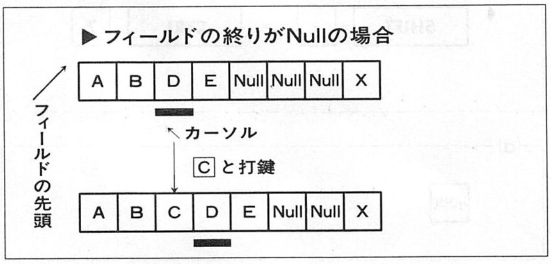
②カーソルの示すアドレスの内容はNullではありませんが、カーソルの示すアドレス以降そのフィールドの終りまでにNullが存在する場合、データ・キーを打鍵するとカーソル位置にその内容が入り、直前にカーソル位置にあった文字、および同一フィールド内のそれ以降の文字は1文字分右へシフトします．ただし、Nullおよびそれ以降の文字はシフトすることはありません．

③カーソル以降そのフィールドの終りまでNullが存在しない場合、挿入動作は行なわれません．

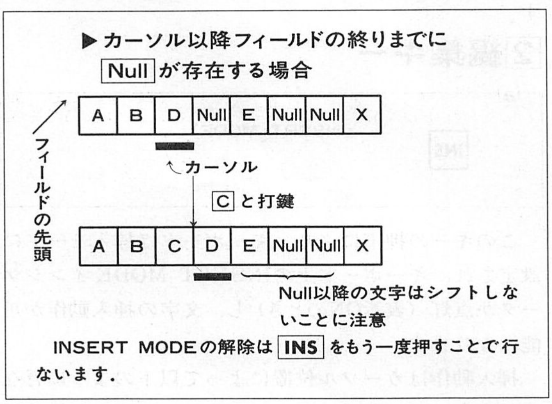
〈例 1〉



〈例 2〉



〈例 3〉



▶コントロール・シフトによる同一機能キー

INS ＝ CTRL − R

(b)

DEL

このキーを押すと、カーソル位置にあった文字は削除され、カーソル以降Nullの直前か、またはフィールドの最後までの文字は1文字分左へシフトし、その後にNullが挿入されます.

(c)

BS

このキーは、カーソル位置がフィールドの先頭でないならば、カーソルを1つ左に移動し、DEL と同じ動作をします.

▶コントロール・シフトによる同一機能キー

BS ＝ CTRL − H

(d)

EL

このキーは、カーソル以降フィールドの終りまでをNullにします.

▶コントロール・シフトによる同一機能キー

EL ＝ CTRL − E

(e)

TAB

このキーはタブ動作が選択されていれば、カーソルを右方向に次のタブ停止位置まで移動させます. また、移動した範囲はスペースになります.

▶コントロール・シフトによる同一機能キー

TAB ＝ CTRL − I

(f)

DUP

このキーは、カーソルが画面の最上段でなければ、同じ桁で1つ上の行の文字を右方向にタブ停止位置まで複写します.

◀コントロール・シフトによる同一機能キー

DUP ＝ CTRL − Q

3 終端キー

(a)

[RETURN]

このキーは、入力を終結するのに使います．通常は正常終了を意味します．

コントロール・シフトによる同一機能キー

[RETURN] ＝ [CTRL]－[M]

(b)

[CTRL]－[C] または [CTRL]－[X]

このキーは、入力を終結するのに使います．通常は取り消しを意味します．

(c)

[CLEAR]

このキーは、現在の入力画面範囲を消去するのに使いますが、プログラムに消去を通知するために入力を終結します．通常はプログラムで再び入力状態にされます．

2.PUTコマンド

1 表示停止

[ESC]

このキーが入力されると現在依頼されている文字列表示後、新たにキー入力されるまでPUTコマンドの状態で待ちます．

停止状態を解除するには、STOPおよびPFを除くいずれのキーであってもかまいません．

(注)コンソール制御でプット・ウェイトが選択されていない場合は、上記動作は行ないません．

2 ページ・リミット

ページ・モード画面に表示中に、画面の最終文字位置に文字を表示するか、または、最終行表示中にＬＦオーダを受け付けた場合、ベルが鳴り、表示動作は停止します．

停止状態を解除するにはSTOPおよびPFを除くいずれかのキーを入力します．

(注)コンソール制御でページ・ウェイトが選択されていない場合は、上記動作は行ないません．

第3章 グラフィック

§1.表示

グラフィック画面は、R、G、Bそれぞれ独立なV-RAM上のビットのON/OFFの合成で作られます．

グラフィック画面は、横640ドット、縦200ドットの大きさです．グラィック画面のアドレス（座標は）横(X)２バイト、縦(Y)２バイトで構成されます．

カラーコード、ファンクション・コードおよびカラー論理演算のカラーコードを示します．

カラーコード表

コード番号	G	R	B	色	補色
0	0	0	0	黒	白
1	0	0	1	青	黄
2	0	1	0	赤	水色
3	0	1	1	紫	緑
4	1	0	0	緑	紫
5	1	0	1	水色	赤
6	1	1	0	黄	青
7	1	1	1	白	黒

ファンクション・コード表

機能名	コード	機能
PSET	0	指定色のドットを表示．
PRESET	1	ドットを背景色にする．
OR	2	出力色と画面色のORをとる．
AND	3	出力色と画面のANDをとる．
XOR	4	出力色と画面色のXORをとる．
NOT	5	出力ビット・パターンをNOTする．

OR演算カラーコード表

出力＼画面	黒 000	青 001	赤 010	紫 011	緑 100	水色 101	黄 110	白 111
黒 000	黒	青	赤	紫	緑	水色	黄	白
青 001	青	青	紫	紫	水色	水色	白	白
赤 010	赤	紫	赤	紫	黄	白	黄	白
紫 011	紫	紫	紫	紫	白	白	白	白
緑 100	緑	水色	黄	白	緑	水色	黄	白
水色 101	水色	水色	白	白	水色	水色	白	白
黄 110	黄	白	黄	白	黄	白	黄	白
白 111	白	白	白	白	白	白	白	白

AND演算カラーコード表

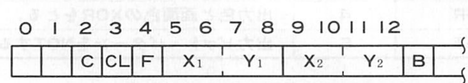
出力 \ 画面	黒 000	青 001	赤 010	紫 011	緑 100	水色 101	黄 110	白 111
黒 000	黒	黒	黒	黒	黒	黒	黒	黒
青 001	黒	青	黒	青	黒	青	黒	青
赤 010	黒	黒	赤	赤	黒	黒	赤	赤
紫 011	黒	青	赤	紫	黒	青	赤	紫
緑 100	黒	黒	黒	黒	緑	緑	緑	緑
水色 101	黒	青	黒	青	緑	水色	緑	水色
黄 110	黒	黒	赤	赤	緑	緑	黄	黄
白 111	黒	青	赤	紫	緑	水色	黄	白

XOR演算カラーコード表

出力 \ 画面	黒 000	青 001	赤 010	紫 011	緑 100	水色 101	黄 110	白 111
黒 000	黒	青	赤	紫	緑	水色	黄	白
青 001	青	黒	紫	赤	水色	緑	白	黄
赤 010	赤	紫	黒	青	黄	白	緑	水色
紫 011	紫	赤	青	黒	白	黄	水色	緑
緑 100	緑	水色	黄	白	黒	青	赤	紫
水色 101	水色	緑	白	黄	青	黒	紫	赤
黄 110	黄	白	緑	水色	赤	紫	黒	青
白 111	白	黄	水色	緑	紫	赤	青	黒

§2.グラフィック・コマンド

1 LINE

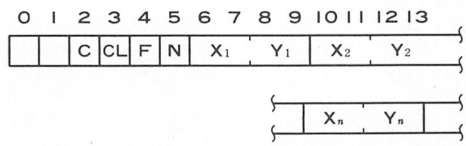


●**機能**：X_1・Y_1からX_2・Y_2まで直線を引くか、または四角を描きます．

- C：コマンド・コード　15_{16}　(21)
- CL：カラーコード
- F：ファンクション・コード（NOTを除く）
- X_1，Y_1：座標1　2Bytes 2進
- X_2，Y_2：座標2　2Bytes 2進
- B：ボックス・フラグ（0：直線　1：四角形　2：四角に塗る）

●**復帰情報**：なし

2 CHAIN

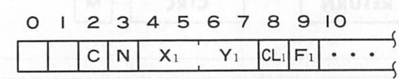


●**機能**：指定座標間を直線で結びます．

- C：コマンド・コード　16_{16}　(22)
- CL：カラーコード
- F：ファンクション・コード（NOTを除く）
- N：座標点数（最大30）
- X_n，Y_n：座標

●**復帰情報**：なし

3 POINT

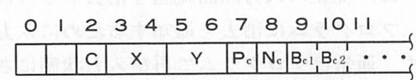


●**機能**：n個の点を表示します．

- C：コマンド・コード　17_{16}　(23)
- N：表示点数（最大20個）
- Xn，Yn：点座標
- CLn：カラーコード
- Fn：ファンクション・コード(NOTを除く)

●**復帰情報**：なし

4 PAINT



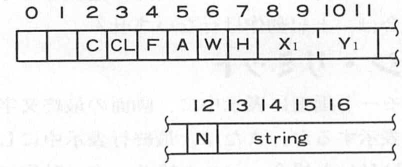
●**機能**：初期座標の位置から指定された境界色の内部を、指定色で塗ります．

- C：コマンド・コード　18_{16}　(24)
- X，Y：初期座標
- Pc：ペイント・カラーコード
- Nc：境界色の数　max 8
- Bcn：境界色

●**復帰情報**：なし

(注)ペイント・カラーおよび画面の端は、境界色とみなします．

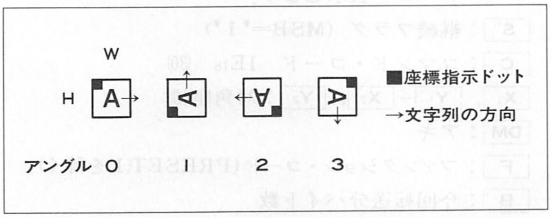
5 SYMBOL



●**機能**：文字列を指定座標より指定角度、大きさで表示します．この場合文字を構成するドット以外は変化しません．

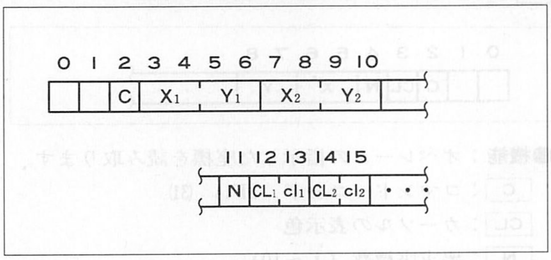
- C：コマンド・コード　19_{16}　(25)

CL ：カラーコード
F ：ファンクション・コード
A ：アングル・コード　0、1、2、3（90°ずつ）
W ：文字横幅倍率（1で8ドット）
H ：文字高さ倍率（1で8ドット）
X1 , Y1 ：座標
N ：文字数（最大80）
String ：JISコード文字列



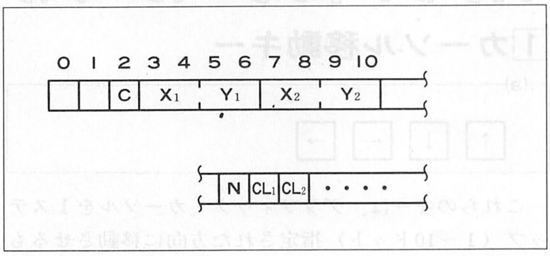
●復帰情報：なし

6 CHANGE COLOR

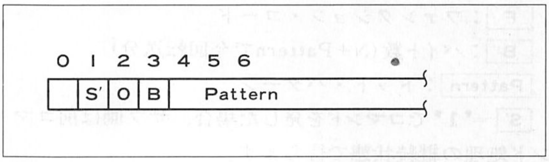


●機能：対角線で示される枠内の色を変えます．
C ：コマンド・コード　1A16 (26)
X1 , Y1 － X2 , Y2 ＝対角線座標
N ：変更数（最大8）
CLn ：旧カラーコード
cln ：新カラーコード
●復帰情報：なし

7 GET BLOCK 1

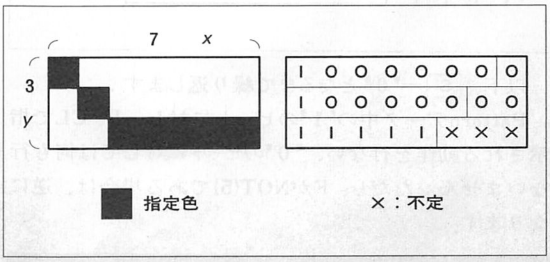


●機能：対角線で示される枠内の指定された色を"1"その他を"0"として画面を読み取ります．
C ：コマンド・コード　1B16 (27)
X1 , Y1 － X2 , Y2 ：対角線座標
N ：指定色数（1～8）
CLn ：カラーコード
●復帰情報

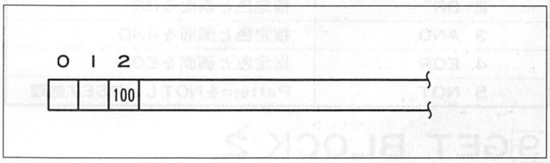


S' ：データ継続フラグ・ビット（MSB）
B ：今回転送パターン・バイト数
Pattern ：画面データ

〈例〉

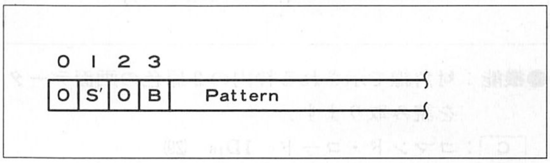


S' ＝"1"の場合、本体側は、次のように残りのデータを要求します．



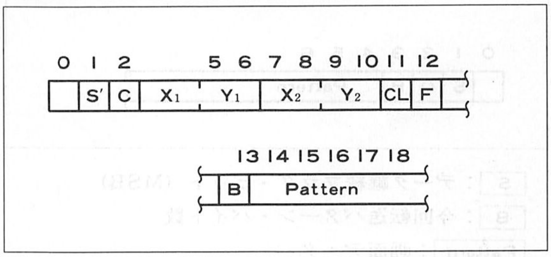
コマンド・コード100は、前コマンド処理継続を示します．

●次の復帰情報



以下、S' ＝"0"となるまで繰り返します．

8 PUT BLOCK 1



●機能：対角線で示す枠内に指定色でパターンを表示します．
S' ：処理継続フラグ・ビット（MSB）
C ：コマンド・コード　1C16 (28)
X1 , Y1 － X2 , Y2 ：対角線座標
CL ：カラーコード

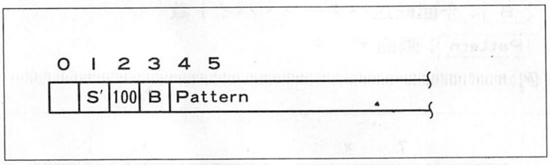
F ：ファンクション・コード

B ：バイト数(N+Patternで今回転送分)

Pattern ：ドット・パターン

S' ＝'1'でコマンドを発した場合、サブ側は前コマンド処理の継続状態で待ちます．

本体側は次のように残りのデータを渡します．

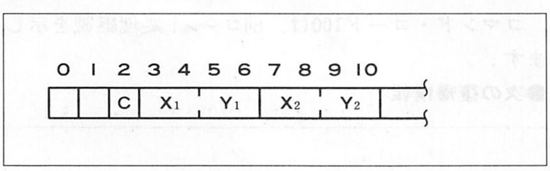


以下、S' ＝'0'となるまで繰り返します．

Patternデータ中の'1'のビットに対し、F、CLで指示される動作を行ない、'0'のビットに対しては何も行ないません．ただし、FがNOT(5)である場合は、逆になります．

ファンクション・コード	動　　作
0 PSET	指定色ドットを表示
1 PRESET	背景色ドットを表示
2 OR	指定色と画面をOR
3 AND	指定色と画面をAND
4 EOR	指定色と画面をEOR
5 NOT	PatternをNOTし、PSET処理

9 GET BLOCK 2

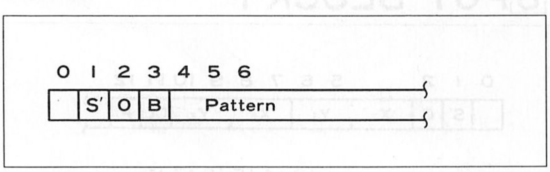


●機能：対角線で示される枠内の３原色の画面データを読み取ります．

C ：コマンド・コード　$1D_{16}$　(29)

X1 , Y1 － X2 , Y2 ：対角線座標

●復帰情報



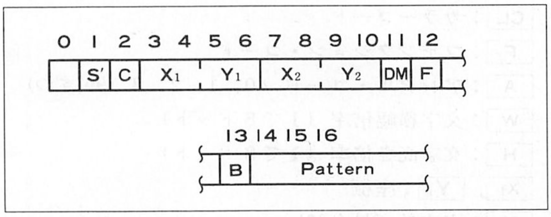
S' ：データ継続フラグ・ビット（MSB）

B ：今回転送パターン・バイト数

Pattern ：画面データ

S' ＝'1'の場合の処理は、7 GET BLOCK 1と同じで、青、赤、緑の順です．

10 PUT BLOCK 2



●機能：対角線で示す枠内にGET BLOCK2で読んだデータを表示します．

S' ：継続フラグ（MSB＝'1'）

C ：コマンド・コード　$1E_{16}$　(30)

X1 , Y1 － X2 , Y2 ：対角座標

DM ：アキ

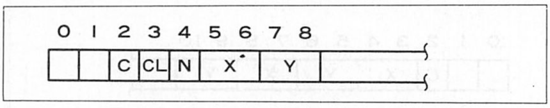
F ：ファンクション・コード(PRESET(1)を除く)

B ：今回転送分バイト数

Pattern ：出力画面データ

S' ＝'1'の処理は、8 PUT BLOCK 1と同じ．

11 GRAPHIC CURSOR



●機能：オペレータの指示した座標を読み取ります．

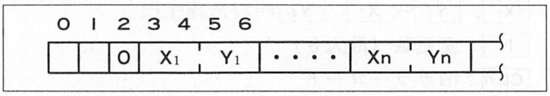
C ：コマンド・コード　$1F_{16}$　(31)

CL ：カーソルの表示色

N ：要求座標数（1～10）

X , Y ：初期座標

●復帰情報

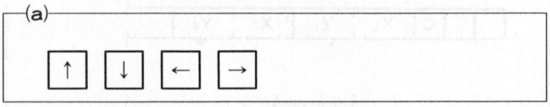


§3.オペレータ制御機能

1.グラフィック・カーソル

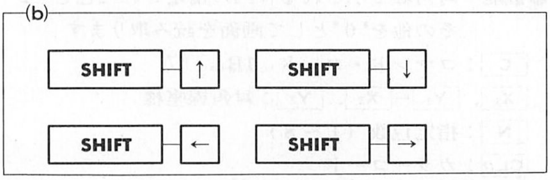
1 カーソル移動キー

(a)



これらのキーは、グラフィック・カーソルを１ステップ（1～10ドット）指定された方向に移動させるものです．

(b)



これらのキーは、グラフィック・カーソルを指定方

向に20ドット単位に移動させるものです．

2

1 ～ 0 (数字キー)

これらのキーは、カーソル1ステップ移動のドット数を設定します．1～9は、その数で0は10を移動ドット数にします．
初期値は1です．

3

RETURN

座標の読み取りを行ないます．指示数の座標が読み取られたら、コマンドを終了します．

第4章キーボード

STOPキーを除くキーは、すべてサブシステムで読み取られます．

PFキーを除くキー(すべてのSHIFTモード)は8ビットのキーコードで表わされます　キーコードについては、ハードウェアの解説書を参照してください．ここではPFキーの扱いについて説明します．

PF(Programable Function) KEY

サブシステムは、10個のPFを持ち、1～10の番号で識別します．PFは、個々に次の2つの機能のうち、いずれか一方を選択できます．

(a)文字列の生成

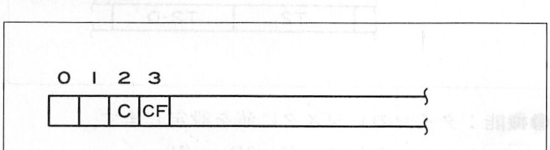
PFキーを押すことにより、複数のデータ・キーが連続して押されたのと同等の機能を持ちます．

(b)割り込みの発生

PFキーを押すことにより、メイン・システムにPF割り込みを起こす．また、この場合COMMUNICATION BUFFER中のSTATUSバイトの0～3ビットにPF番号が置かれます．

§1.キーボード・コマンド

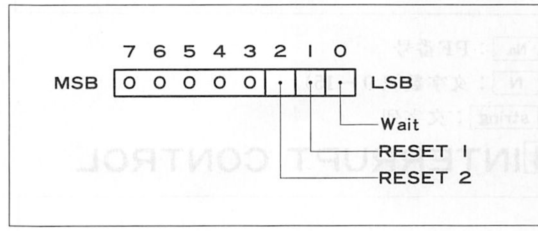
1 INKEY



●**機能**：キーコードを読み取ります．

C：コマンド・コード　29₁₆　(41)

CF：制御フラグ



RESET 1、2のいずれかがONの場合Read動作に先だって次のことが行なわれます．

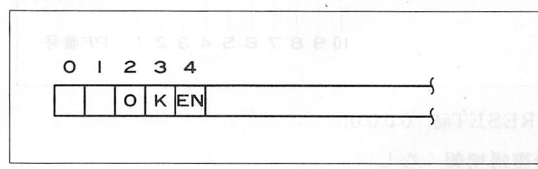
RESET1=′1′：KEYレジスタを何も入力されていない状態にします．

RESET2=′1′：KEYレジスタの内容がPF生成コードでないならばRESET1=′1′の動作をします．

同時にONならRESET1=′1′の動作を取ります

Wait=′1′のときは、キーの入力を待ちます．

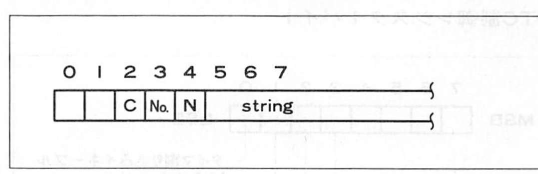
●**復帰情報**



K：キーコード

EN：入力の有無　0：無
　　　　　　　　1：有

2 DEFINE STRING OF PF



●**機能**：ファンクション・キーに文字列を定義します．

C：コマンド・コード　2A₁₆　(42)

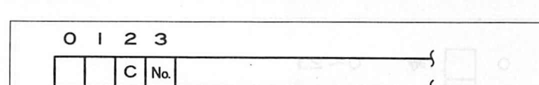
No.：ファンクション・キー番号　1～10

N：文字数　0～15

string：文字列

●**復帰情報**：なし

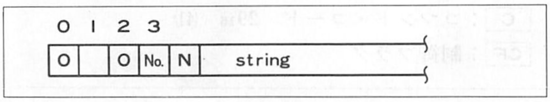
3 GET STRING OF PF



●**機能**：PFに定義されている文字列を読み取ります．

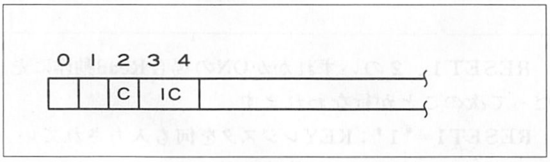
C：コマンド・コード　2B₁₆　(43)

No.：PF番号（1～10）

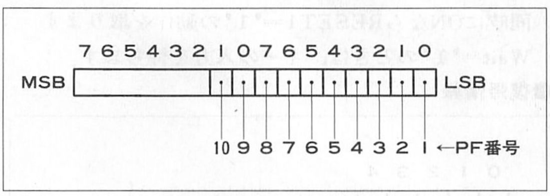


No.：PF番号
N：文字数（0～15）
string：文字列

4 INTERRUPT CONTROL



●**機能**：PF割り込みの選択を行ないます.
C：コマンド・コード　2C16　(44)
IC：割り込み制御　'1'で割り込み



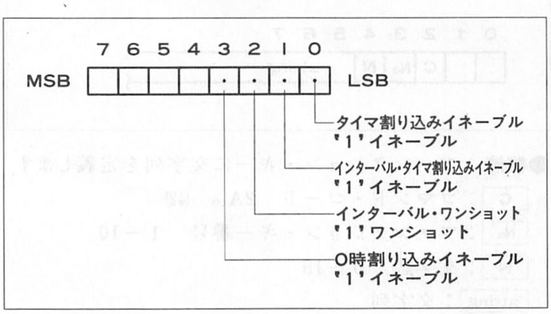
RESET時　0000₁₆

●**復帰情報**：なし

第5章 タイマ

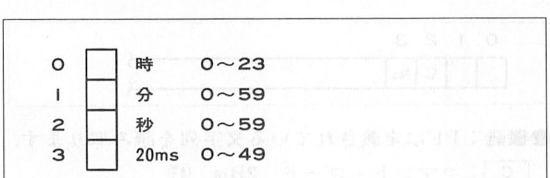
タイマは、次の5つのレジスタで構成されます.

①**TC制御レジスタ1バイト**



RESET時　0 0₁₆

②**T 1 24時時計4バイト**



③**T 1 -I 割り込み予約時刻**

数値の形式は、T 1と同じ

④**T2　20msデクリメント・カウンタ4バイト**

⑤**T2-D 再設定値レジスタ　4バイト**

◇　◇　◇

タイマー動作を機能別に説明します.

①**時計**

T1は、20msに1回カウント・アップし、自動的に秒、分、時の桁上りを行ないます.

②**タイマ割り込み**

T1が、カウント・アップされたときに、TCのBit 0が'1'であるならば、T1とT1-Iの値を比較し等しければ*タイマ割り込みフラグを設定し、割り込みをかけます.

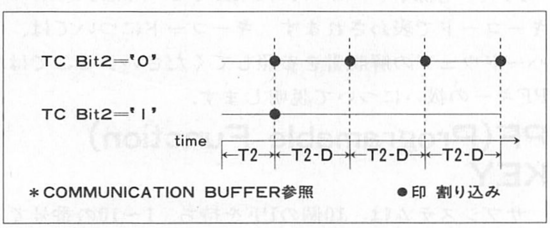
③**カウンタ**

T2は、20msに1回カウント・ダウンし、結果が0であればT2-Dの値をT2にロードします(RESTORE).

④**インターバル・タイマ割り込み**

RESTOREを行なったときにTCのBit1が'1'であるならば、*インターバル・タイマ割り込みフラグを設定し、割り込みをかけます.

この後、TCのBit2が'1'ならば、TCのBit 1を'0'にします. このことは2つのモードを作ることになります.



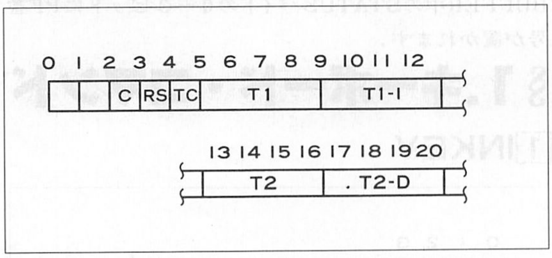
⑤**0時割り込み**

T1がカウント・アップされたときに、T1のすべての桁が0で、TCのBit 3が'1'ならば、0時刻割り込みフラグを設定し、割り込みをかけます.

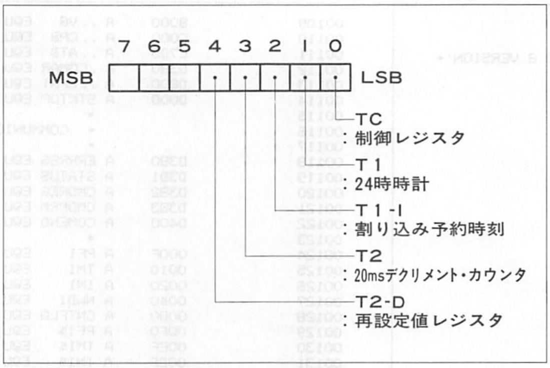
◇　◇　◇

タイマ・コマンド

1 SET TIMER

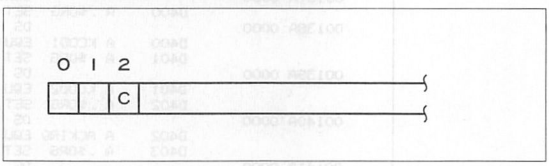


●**機能**：タイマのレジスタに値を設定します.
C：コマンド・コード　3D16　(61)
RS：設定するレジスタの選択（ビット・オン）



●復帰情報：なし

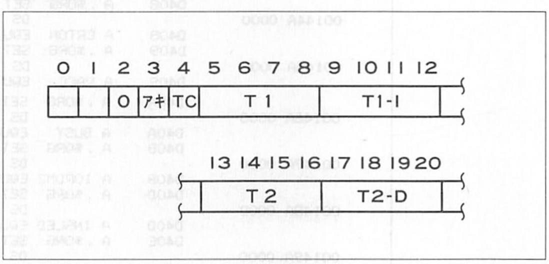
2 READ TIMER



●機能：タイマのレジスタを読み取ります．

C：コマンド・コード 3E16 (62)

●復帰情報



TC：制御レジスタ

T1：24時時計

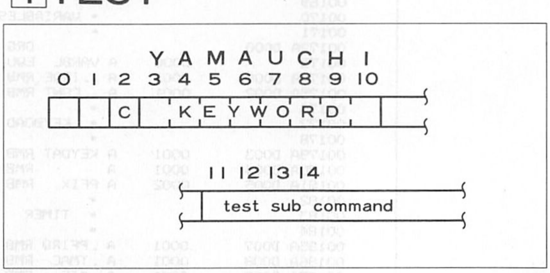
T1-I：割り込み予約時刻

T2：20msデクリメント・カウンタ

T2-D：再設定値レジスタ

第6章 メンテナンス・コマンド

1 TEST



●機能：種々のサブコマンドにより、サブシステム内を絶対アドレスで参照できる．

C：コマンド・コード 3F16 (63)

KEYWORD：キーワード（YAMAUCHI）

8バイトすべてが、システムで決定されたものと一致しない場合は、このコマンドは、動作しません．

test sub command：サブコマンドの並び

§ 1.サブコマンド

サブコマンドは、サブコマンド・コードと複数個のオペランド（ない場合もある）で構成されます．

未定義のサブコマンド・コードの場合は、TESTコマンドを終了します．

1 ENDサブコマンド

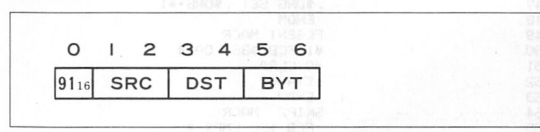
●形式：



●機能：TESTコマンドの実行を終了します．

2 MOVEサブコマンド

●形式：



●機能：SRCで示されるアドレスからDSTで示されるアドレスにBYTのバイト数データを複写します．

(注)BYTを0とした場合、65536バイトのデータを複写します．

3 JMPサブコマンド

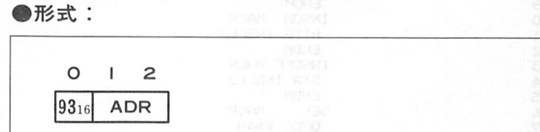
●形式：



●機能：ADRで示されるアドレスへサブコマンドの制御を移します．

4 JMSRサブコマンド

●形式：



●機能：ADRで示されるアドレスの機械語サブルーチンを呼び出します．

機械語ルーチンからの復帰はRTS(39_{16})で行ないます．

サブシステム・モニタ ソース・リスト

```
                        OPT    CAT,MEX
                        OPT    LIST,LLEN=120,CREF
                        TTL    CRT SUB-SYSTEM * FUJITSU MICRO 8 VERSION *
                        NAM    CRTMON
                      *
                      *    CRT SUB-SYSTEM MONITOR
                      *
                      *    FUNCTION BLOCK
                      *      CONSOLE
                      *      GRAPHIC
                      *      KEYBOAD
                      *      TIMER
                      *
                      *    FACILITIES
                      *      DISPLAY(VRAM 16K*3)
                      *      KEYBORD
                      *      20 MS TIMER
                      *      BUZZER
                      *
                      *    DESIGNED BY M.YAMAUCHI
                      *
                      *    FEB. 1981 SUB-SYSTEM2 V0.0
                      *    MAR.  3 1981 SUB SYSTEM2 V0.1
                      *    APR.  3 1981 V0.2
                      *    APR.  10 1981 V0.3
                      *    APR.  16 1981 V1.0
                      *    APR.  30 1981 V1.1
                      *    JUNE 3 1981 V3.0
                      *    JUNE 6 1981 V3.1
                      *    JUNE 10 1981 V3.2
                      *
                      *    JULY 7 1981 V3.3 FM8 HIGH SPEED
                      *                     VERSION
                      *
                      *    SEP.  3 1981 V4.0 MASK ROM VERSION
                      *
                      *    CODED BY M.YAMAUCHI
                      *
                      *    ALL RIGHT RESEAVED
                      *
                      SORG   MACR
                      .$ORG SET ¥0
                       ENDM
                      DS     MACR
                      ¥0 EQU .$ORG
                      .$ORG SET .$ORG+¥1
                       ENDM
                      FLGENT MACR
                      ¥1 FCB $86 LDA #
                      ¥0 CLRA
                       STA ¥2
                       ENDM
                      SKIP2  MACR
                       FCB $8C CMPX #
                       ENDM
                      SKIP1  MACR
                       FCB $21
                       ENDM
                      RTSPUL MACR
                       PULS ¥0,PC
                       ENDM
                      ACK    MACR
                       BITA ACKIRQ
                       ENDM
                      SCRON  MACR
                       BITA CRTON
                       ENDM
                      SCROFF MACR
                       STA CRTON
                       ENDM
                      VACON  MACR
                       BITA VACC
                       ENDM
                      VACOFF MACR
                       STA VACC
                       ENDM
                      BSYON  MACR
                       STA BUSY
                       ENDM
                      BSYOFF MACR
                       TST BUSY
                       ENDM
                      BUZZON MACR
                       BITA BUZZER
                       ENDM
                      SYSREQ MACR
                       BITA ATTENT
                       ENDM
                      INSON  MACR
                       BITA INSLED
                       ENDM
                      INSOFF MACR
                       STA INSLED
                       ENDM
                      SEF    MACR
                       ORCC #$40
                       ENDM
                      CLF    MACR
                       ANDCC #$BF
                       ENDM
                      *
                      *
                      *    CONSTANT VALUE ON LABEL
                      *
             0000     A ..VB    EQU    $0000    BLUE VRAM
             4000     A ..VR    EQU    $4000    RED VRAM
             8000     A ..VG    EQU
             C000     A ..CRB   EQU
             C7D0     A ..ATB   EQU
             D380     A .COMAR  EQU
             D800     A ..CFNT  EQU
             D000     A STKTOP  EQU
                        *
                        *  COMMUNICAT
                        *
             D380     A ERRREG  EQU
             D381     A STATUS  EQU
             D382     A CMDREG  EQU
             D383     A CMDPRM  EQU
             D400     A COMEND  EQU
                        *
             000F     A PFI     EQU
             0010     A TMI     EQU
             0020     A INI     EQU
             0040     A NWDI    EQU
             0080     A CNTFLG  EQU
             00F0     A PFI$    EQU
             00EF     A TMI$    EQU
             00DF     A INI$    EQU
             00BF     A NWDI$   EQU
             007F     A CNTFL$  EQU
                        *
                        *  I/O REGIST
                        *
A 0000                          SORG
             D400     A .$ORG   SET
A 0000                          DS
             D400     A KCOD1   EQU
             D401     A .$ORG   SET
A 0000                          DS
             D401     A KCOD2   EQU
             D402     A .$ORG   SET
A 0000                          DS
             D402     A ACKIRQ  EQU
             D403     A .$ORG   SET
A 0000                          DS
             D403     A BUZZER  EQU
             D404     A .$ORG   SET
A 0000                          DS
             D404     A ATTENT  EQU
             D405     A .$ORG   SET
A 0000                          DS
             D405     A IORDUM  EQU
             D408     A .$ORG   SET
A 0000                          DS
             D408     A CRTON   EQU
             D409     A .$ORG   SET
A 0000                          DS
             D409     A VACC    EQU
             D40A     A .$ORG   SET
A 0000                          DS
             D40A     A BUSY    EQU
             D40B     A .$ORG   SET
A 0000                          DS
             D40B     A IORDM2  EQU
             D40D     A .$ORG   SET
A 0000                          DS
             D40D     A INSLED  EQU
             D40E     A .$ORG   SET
A 0000                          DS
             D40E     A SOFFST  EQU
             D410     A .$ORG   SET
                        *
                        *
             0007     A AMCL    EQU
             0008     A AMRV    EQU
             0010     A AMPT    EQU
             0040     A AMMD    EQU
             0080     A AMFL    EQU
                        *
             0050     A CSCX1   EQU
             0028     A CSCX2   EQU
             0019     A CSCY1   EQU
             0014     A CSCY2   EQU
             0050     A GXBSZ   EQU
             0280     A GXD     EQU
             00C8     A GYD     EQU
             0008     A CFH     EQU
                        *
             0010     A PFSZ    EQU
                        *
                        * VARIABLES
                        *
A D000                          ORG
             D000     A VARBL   EQU
A D000       0002     A ..IRQE  RMB
A D002       0001     A ..CONT  RMB
                        *
                        *  KEYBOAD
                        *
A D003       0001     A KEYDAT  RMB
A D004       0001     A         RMB
A D005       0002     A PFIX    RMB
                        *
                        *  TIMER
                        *
A D007       0001     A .PFIRQ  RMB
A D008       0001     A .TMAC   RMB
A D009       0001     A .$TC    RMB
A D00A       0004     A .$T1    RMB
A D00E       0004     A .$T1I   RMB
A D012       0004     A .$T2    RMB
A D016       0004     A .$T2D   RMB
```

```
$8000   GREEN VRAM
$C000   CHARACTER BUFFER
..CRB+2000 ATTRIBUTE BUFFER
$D380   COMMUNICATION AREA
$DB00   ORIGINAL FONT
$D000

ION REG.

.COMAR  ERROR CODE
.COMAR+1
.COMAR+2 COMMAND CODE
.COMAR+3 PARAMETER
.COMAR+128 END OF AREA PAST 1
$0F       STATUS 0-3
$10       STATUS 4
$20       STATUS 5
$40       STATUS 6
$80       STATUS 7
-PFI-$FF01
-TMI-$FF01
-INI-$FF01
-NWDI-$FF01
-CNTFLG-$FF01

OR

$D400
$D400
KCOD1,1 MSB KDB
.$ORG
.$ORG+1
KCOD2,1 KDO-KD7
.$ORG
.$ORG+1
ACKIRQ,1 ACKNOWLEGE IRQ
.$ORG
.$ORG+1
BUZZER,1 BUZZER
.$ORG
.$ORG+1
ATTENT,1 MAIN SYSTEM REQUEST
.$ORG
.$ORG+1
IORDUM,3 DUMMY
.$ORG
.$ORG+3
CRTON,1 DISPLAY ON/OFF
.$ORG
.$ORG+1
VACC,1  VRAM ACCESS
.$ORG
.$ORG+1
BUSY,1  BUSY STATE
.$ORG
.$ORG+1
IORDM2,2 DUMMY
.$ORG
.$ORG+2
INSLED,1 INSERT MODE LED
.$ORG
.$ORG+1
SOFFST,2 VRAM OFFSET REG.
.$ORG
.$ORG+2


$07      COLOR MASK ON ATTR.
$08      REVERSE MASK ON ATTR.
$10      PROTECT MASK ON ATTR.
$40      MODIFYED TAG MASK ON ATTR.
$80      FIELD START MASK ON ATTR.

80       NUMBER OF COLUMN IN LINE,MANY
40       NUMBER OF COLUMN IN LINE,LITTLE
25       NUMBER OF LINE,MANY
20       NUMBER OF LINE,LITTLE
80       HORIZONTAL BYTES,GRAPHIC
640      HORIZONTAL DOT
200      VERTICAL DOT
8        HEIGHT OF CHAR FONT

16


$D000
*
2        IRQ ENTRY ADDRESS
1        CONT FLAG SAVE



1        FLAG
1        CODE
2        PF STRING PTR



1        PF INTERRUPT CODE
1        ACCESS FLAG
1        CONTROL
4        CLOCK
4        INTERRUPT TIME
4        COUNTER
4        RESTORE DATA
```

```
00192                                *
00193A D01A          0001     A .CURSR RMB    1         STATE REG. OF CURSOR DISPLAY
00194A D01B          0001     A .CURTM RMB    1         TIMER OF CURSOR
00195A D01C          0001     A .CURST RMB    1
00196A D01D          0002     A .VOFST RMB    2         VRAM OFFSET FOR SCROLL
00197A D01F          0001     A .CSCTL RMB    1
00198A D020          0001     A PUTCFG RMB    1         CONTINUANCE PUT FLAG.
00199A D021          0001     A CSLCDS RMB    1         CONSOLE COLOR DISABLE(MONO SELECTION)
00200A D022          000A     A .TABST RMB    10
00201A D02C          0001     A ..ACL  RMB    1         ATTR COLOR CODE
00202A D02D          0001     A ..BCL  RMB    1         BACK COLOR CODE
00203A D02E          0001     A ..CLEX RMB    1
00204A D02F          0001     A ..CWD  RMB    1         CHAR WIDTH FLAG,0=SMALL,1=LARGE
00205A D030          0001     A ..CHT  RMB    1         CHAR UNDER MARGIN,0 OR 2.
00206A D031          0001     A ..INS  RMB    1         INSERT FLAG
00207A D032          0001     A ..EKEY RMB    1         ENTER KEY CODE
00208A D033          0001     A ..LNCD RMB    1         0 LONG
00209A D034          0001     A ..LNC  RMB    1         NUMBER OF COLUMNS IN LINE
00210A D035          0001     A ..DNL  RMB    1         NUMBER OF LINES IN DISPLAY
00211A D036          0001     A ..FS   RMB    1         KEY INDICATE FLAG
00212A D037          0002     A ..VCU  RMB    2         VRAM ADDR UNIT OF CHR COLUMN
00213A D039          0002     A ..VLU  RMB    2         VRAM ADDR UNIT OF CHR LINE
00214A D03B          0008     A ..DMAP RMB    8         MAP OF DISPLAY PLACE
00215A D043          0001     A .BPY   RMB    1         BUFFER PTR Y
00216A D044          0001     A .BPX   RMB    1         BUFFER PTR X
00217A D045          0002     A .BPAD  RMB    2         BUFFER PTR (ADDR)
00218A D047          0002     A .BAAD  RMB    2         BUFFER PTR,ATTR.
00219A D049          0002     A .BVAD  RMB    2         BUFFER PTR,VRAM ADDR
00220A D04B          0003     A .MDRG  RMB    3         CURR. DISPLAY WINDOW
00221A D04E          0001     A .CY    RMB    1         CUSOR ADDR Y
00222A D04F          0001     A .CX    RMB    1         CUSOR ADDR X
00223A D050          0002     A .CBAD  RMB    2         CUSOR BUFFER ADDR
00224A D052          0002     A .CAAD  RMB    2         CUSOR ATTR ADDR
00225A D054          0002     A .CVAD  RMB    2         CUSOR VRAM ADDR
00226A D056          0001     A .GCL   RMB    1         COLOR CODE
00227A D057          0001     A .GF    RMB    1         FUNCTION CODE
00228A D058          0002     A .GY    RMB    2         AXIS Y GRAPH
00229A D05A          0002     A .GX    RMB    2         AXIS X GRAPH
00230A D05C          0002     A .GAD   RMB    2         VRAM ADDR
00231A D05E          0001     A .GBMK  RMB    1         BIT MARK
00232A D05F          0002     A .BOXX1 RMB    2         BOX X1
00233A D061          0002     A .BOXY1 RMB    2         BOX Y1
00234A D063          0002     A .BOXX2 RMB    2         BOX X2
00235A D065          0002     A .BOXY2 RMB    2         BOX Y2
00236A D067          0002     A .LX1   RMB    2         X CO-ORDINATE OF POINT1
00237A D069          0002     A .LY1   RMB    2         Y CO-ORDINATE OF POINT1
00238A D06B          0002     A .LX2   RMB    2         X CO-ORDINATE OF POINT2
00239A D06D          0002     A .LY2   RMB    2         Y CO-ORDINATE OF POINT2
00240A D06F          0002     A .CBTOP RMB    2         CLEAR BUFFER TOP ADDR
00241A D071          0002     A .CBBTM RMB    2         CLEAR BUFFER BOTTOM ADDR+1
00242A D073          0002     A .CVTOP RMB    2         CLEAR VRAM TOP ADDR
00243A D075          0002     A .CVBTM RMB    2         CLEAR VRAM BOTTOM ADDR+1
00244A D077          0002     A .ERSIX RMB    2         ERASE IX
00245                                *
00246A D079                            SORG   .CBTOP
                     D06F     A .$ORG  SET    .CBTOP
00247A D079                            DS     BLUSID,3 LINEX SIDE. BLUE
                     D06F     A BLUSID EQU    .$ORG
                     D072     A .$ORG  SET    .$ORG+3
00248A D079                            DS     REDSID,3 LINEX SIDE. RED
                     D072     A REDSID EQU    .$ORG
                     D075     A .$ORG  SET    .$ORG+3
00249A D079                            DS     GRESID,3 LINEX SIDE. GREEN
                     D075     A GRESID EQU    .$ORG
                     D078     A .$ORG  SET    .$ORG+3
00250                                *
00251A D079          0012     A EXFONT RMB    18        EXTD FONT
00252A D08B                            SORG   EXFONT
                     D079     A .$ORG  SET    EXFONT
00253A D08B                            DS     BLUDSP,3 RAM HOOK BLUE DOT DISPLAY
                     D079     A BLUDSP EQU    .$ORG
                     D07C     A .$ORG  SET    .$ORG+3
00254A D08B                            DS     REDDSP,3 RAM HOOK RED DOT DISPLAY
                     D07C     A REDDSP EQU    .$ORG
                     D07F     A .$ORG  SET    .$ORG+3
00255A D08B                            DS     GREDSP,3 RAM HOOK FOR GREEN DOT DISPLAY
                     D07F     A GREDSP EQU    .$ORG
                     D082     A .$ORG  SET    .$ORG+3
00256A D08B                            DS     BLULIN,3 RAM HOOK FOR X PARALLEL LINE'S BLUE
                     D082     A BLULIN EQU    .$ORG
                     D085     A .$ORG  SET    .$ORG+3
00257A D08B                            DS     REDLIN,3 RAM HOOK FOR X PARALLEL LINE'S RED
                     D085     A REDLIN EQU    .$ORG
                     D088     A .$ORG  SET    .$ORG+3
00258A D08B                            DS     GRELIN,3 RAM HOOK FOR X PARALLEL LINE'S GREEN
                     D088     A GRELIN EQU    .$ORG
                     D08B     A .$ORG  SET    .$ORG+3
00259A D08B          0002     A HKFLG  RMB    2         CURRENT COLOR FOR CHARACTER
00260A D08D          0003     A HKBLUE RMB    3         BLUE
00261A D090          0003     A HKRED  RMB    3         RED
00262A D093          0003     A HKGREE RMB    3         GREEN
00263                                *
00264A D096          022A     A .WORK  RMB    .COMAR-(PFSZ*10+$20)-*
00265                                *
00266A D2C0          00A0     A .PFTBL RMB    PFSZ*10 PF TABLE
00267                D360     A VARBL$ EQU    *
00268                                *
00269                                ****************************************
00270                                  TTL    MAIN - CRT SUBSYYSTEM2 -
00272                00D0     A        SETDP  $D0
00273A E000                            ORG    $E000
00274                                *
00275                                * R E S T A R T   P O I N T
00276                                *
00277                E000     A RESTAR EQU    *
00278A E000 86       D0       A        LDA    #$D0
00279A E002 1F       BB       A        TFR    A,DP
00280A E004 CC       0000     A        LDD    #0        INIT VARIABLE
```

```
00281A E007 8E   D000    A        LDX    #VARBL
00282A E00A ED   81      A RESTOO STD    ,X++
00283A E00C 8C   D360    A        CMPX   #VARBL$
00284A E00F 25   F9   E00A        BLO    RESTOO
00285A E011 86   07      A        LDA    #$07
00286A E013 97   2C      A        STA    ..ACL  ATTRIBUTE
00287A E015 FC   D400    A        LDD    KCOD1  TURN OFF THE KEY STROBE.
00288                    *
00289A E018 10CE D000    A        LDS    #STKTOP INIT SP
00290                    *
00291A E01C BD   E1D8    A        JSR    DEFALT DEFAULT INIT
00292A E01F 86   23      A        LDA    #$23
00293A E021 BD   ED77    A        JSR    CNSINT
00294                    *
00295A E024 7F   D381    A        CLR    STATUS CLEAR INTERRUPT
00296                    *
00297            E027    A CMDCTL EQU    *
00298                    *
00299                    * C O M M A N D   H A N D L E R
00300                    *
00301A E027 10CE D000    A        LDS    #STKTOP INIT SP
00302A E02B                       CLF    CLEAR  KEY MASK
     A E02B 1C   BF      A        ANDCC  #$BF
00303A E02D 7F   D382    A        CLR    CMDREG COMMAND NON
00304A E030 BD   ED94    A        JSR    CURSET
00305A E033 BD   EDB7    A        JSR    CURGO
00306A E036 BD   E12F    A        JSR    CMDWAI
00307                    *
00308            E039    A CMDCT1 EQU    *
00309A E039 34   02      A        PSHS   A
00310A E03B BD   EDB1    A        JSR    CURSTP
00311A E03E BD   E123    A CMDCNT JSR    COMINT CTL REG CLEAR
00312A E041 35   02      A        PULS   A
00313A E043 F6   E06F    A        LDB    CMDPN  NO. OF AVAILABLE COMMAND
00314A E046 8E   E070    A        LDX    #CMDPTR COMMAND ENTRY TABLE
00315                    *
00316A E049 A1   84      A CMDCT2 CMPA   0,X
00317A E04B 25   04   E051        BLO    CMDCT3 LOW
00318A E04D A1   01      A        CMPA   1,X
00319A E04F 23   08   E059        BLS    CMDJMP EXIST THIS BLOCK
00320A E051 30   04      A CMDCT3 LEAX   4,X    NEXT BLOCK
00321A E053 5A                    DECB
00322A E054 26   F3   E049        BNE    CMDCT2
00323A E056 7E   E0E6    A        JMP    ERR70  UNAVAIRABLE
00324                    *
00325            E059    A CMDJMP EQU    *
00326A E059 A0   84      A        SUBA   0,X    OFFSET ON BLOCK
00327A E05B 48                    LSLA
00328A E05C AE   02      A        LDX    2,X    VECTOR TABLE TOP
00329A E05E AE   86      A        LDX    A,X
00330                    *
00331A E060 AD   84      A GOCMD  JSR    ,X     BRANCH TO COMMAND
00332                    *
00333A E062 20   C3   E027 CMSEND BRA    CMDCTL  SEND & WAIT
00334                    *
00335            E064    A IRQAB  EQU    *
00336A E064                       ACK
     A E064 B5   D402    A        BITA   ACKIRQ
00337A E067 9E   00      A        LDX    ..IRQE
00338A E069 27   BC   E027        BEQ    CMDCTL NO ENTRY
00339A E06B AD   84      A        JSR    ,X
00340A E06D 20   B8   E027        BRA    CMDCTL
00341                    *
00342A E06F      04      A CMDPN  FCB    (CON$-CMDPTR)/4
00343A E070      010D    A CMDPTR FDB    $010D,CON$ CONSOLE BLOCK
     A E072      E080    A
00344A E074      1520    A        FDB    $1520,GRA$ GRAPHIC BLOCK
     A E076      E09A    A
00345A E078      292C    A        FDB    $292C,KEY$ KEYBORD BLOCK
     A E07A      E0B2    A
00346A E07C      3D3F    A        FDB    $3D3F,TIM$ TIMER BLOCK
     A E07E      E0BA    A
00347A E080      E15B    A CON$   FDB    INIT
00348A E082      E250    A        FDB    ERASE
00349A E084      E5E5    A        FDB    PUT
00350A E086      E5E4    A        FDB    GET
00351A E088      EBEF    A        FDB    GETC
00352A E08A      ECA1    A        FDB    CGBLK1
00353A E08C      ED1E    A        FDB    CPBLK1
00354A E08E      ECA0    A        FDB    CGBLK2
00355A E090      ED1D    A        FDB    CPBLK2
00356A E092      EC98    A        FDB    GETBAD
00357A E094      ED68    A        FDB    TABSET
00358A E096      ED74    A        FDB    CNSCTL
00359A E098      E246    A        FDB    NERASE
00360A E09A      EE48    A GRA$   FDB    LINE
00361A E09C      F074    A        FDB    CHAIN
00362A E09E      F3F3    A        FDB    POINT
00363A E0A0      FA8B    A        FDB    PAINT
00364A E0A2      F0B3    A        FDB    SYMBOL
00365A E0A4      F1CB    A        FDB    CHANGE
00366A E0A6      F236    A        FDB    GBLK1
00367A E0A8      F427    A        FDB    PBLK1
00368A E0AA      F397    A        FDB    GBLK2
00369A E0AC      F56A    A        FDB    PBLK2
00370A E0AE      F65E    A        FDB    GCURS
00371A E0B0      EE22    A        FDB    CHRLN
00372A E0B2      FD76    A KEY$   FDB    INKEY
00373A E0B4      FDE5    A        FDB    FUNKEY
00374A E0B6      FE29    A        FDB    GSTRPF
00375A E0B8      FE44    A        FDB    PFINT
00376A E0BA      FE5F    A TIM$   FDB    SETTIM
00377A E0BC      FE87    A        FDB    REDTIM
00378A E0BE      FF52    A        FDB    TEST
00379                    *
00380A E0C0      00      A INPARM FCB    0,80,25,0,25,0,1,0
     A E0C1      50      A
     A E0C2      19      A
     A E0C3      00      A
     A E0C4      19      A
     A E0C5      00      A
     A E0C6      01      A
     A E0C7      00      A
00381                    *
00382                    *
00383                    *
00384                    * E R R O R
00385                    *
00386A E0C8 86   3C      A ERR60  LDA
00387A E0CA                       SKIP2
     A E0CA      8C      A        FCB
00388A E0CB 86   3D      A ERR61  LDA
00389A E0CD                       SKIP2
     A E0CD      8C      A        FCB
00390A E0CE 86   3E      A ERR62  LDA
00391A E0D0                       SKIP2
     A E0D0      8C      A        FCB
00392A E0D1 86   3F      A ERR63  LDA
00393A E0D3                       SKIP2
     A E0D3      8C      A        FCB
00394A E0D4 86   40      A ERR64  LDA
00395A E0D6                       SKIP2
     A E0D6      8C      A        FCB
00396A E0D7 86   41      A ERR65  LDA
00397A E0D9                       SKIP2
     A E0D9      8C      A        FCB
00398A E0DA 86   42      A ERR66  LDA
00399A E0DC                       SKIP2
     A E0DC      8C      A        FCB
00400A E0DD 86   43      A ERR67  LDA
00401A E0DF                       SKIP2
     A E0DF      8C      A        FCB
00402A E0E0 86   44      A ERR68  LDA
00403A E0E2                       SKIP2
     A E0E2      8C      A        FCB
00404A E0E3 86   45      A ERR69  LDA
00405A E0E5                       SKIP2
     A E0E5      8C      A        FCB
00406A E0E6 86   46      A ERR70  LDA
00407                    *
00408            E0E8    A ERROR  EQU
00409A E0E8 B7   D380    A        STA
00410A E0EB 7E   E062    A        JMP
00411                    *
00412                    **************
00414            E0EE    A CONTO  EQU
00415                    *
00416                    * INTERMIDIAT
00417                    *
00418                    *   CONTINUOU
00419                    *   BUSY LINE
00420                    *   AND WAIT
00421                    *
00422                    *   IF NEW CO
00423                    *   BREAK NOW
00424                    *
00425A E0EE 34   02      A        PSHS
00426A E0F0 86   80      A        LDA
00427A E0F2 BA   D381    A        ORA
00428A E0F5 B7   D381    A        STA
00429A E0F8 8D   35   E12F        BSR
00430A E0FA 81   64      A        CMPA
00431A E0FC 26   10   E10E        BNE
00432                    *
00433            E0FE    A CONTO2 EQU
00434A E0FE 7F   D380    A        CLR
00435A E101 86   7F      A        LDA
00436A E103 B4   D381    A        ANDA
00437A E106 B7   D381    A        STA
00438A E109 7F   D382    A        CLR
00439A E10C                       RTSPUL
     A E10C 35   82      A        PULS
00440                    *
00441                    **************
00442A E10E 10CE D000    A CNTABT LDS
00443A E112 34   02      A        PSHS
00444A E114 BD   ED94    A        JSR
00445A E117 7E   E03E    A        JMP
00447            E11A    A CONTI  EQU
00448                    *
00449                    * GET CONTINU
00450                    *
00451                    *   BUSY LINE
00452                    *   WAIT FOR
00453                    *   AND SAVE
00454                    *
00455                    *   IF NEW CO
00456                    *   BREAK NOW
00457                    *
00458A E11A 34   02      A        PSHS
00459A E11C 8D   11   E12F        BSR
00460A E11E 81   64      A        CMPA
00461A E120 26   EC   E10E        BNE
00462A E122                       SKIP2
     A E122      8C      A        FCB
00463                    *
00464A E123 34   02      A COMINT PSHS
00465A E125 B6   D381    A CONTI2 LDA
00466A E128 97   02      A        STA
00467A E12A 20   D2   E0FE        BRA
00468                    *
00469                    **************
00471A E12C 32   62      A CMDWAB LEAS
00472A E12E 3B                    RTI
00473            E12F    A CMDWAI EQU
```

```
 E N T R Y

 #60      INIT PARAM

 $8C      CMPX #
 #61      CO-ORDINATE OF CONSOLE

 $8C      CMPX #
 #62      ORDER SEQ.

 $8C      CMPX #
 #63      CO-ORDINATE OF GRAPHIC

 $8C      CMPX #
 #64      FUNCTION CODE

 $8C      CMPX #
 #65      NO. OF CO-ORDINATE

 $8C      CMPX #
 #66      NO. OF CHARACTER

 $8C      CMPX #
 #67      NO. OF COLOR

 $8C      CMPX #
 #68      PF NO.

 $8C      CMPX #
 #69      INVALID PARAMETER

 $8C      CMPX #
 #70      INVALID COMMAND
 *
 .COMAR   SET ERROR STATUS
 CMSEND   SEND
**********************************
 *

E SEND PROCESS

S FLAG ON,
 LOW.
FOR CONTINUE COMMAND.

MMAND IS NOT CONTINUE,
 COMMAND AND DO NEW COMMAND.

 A
 #CNTFLG
 STATUS
 STATUS
 CMDWAI  WAIT
 #100    CONTINUE ?
 CNTABT

 *
 ERRREG
 #CNTFL$
 STATUS
 STATUS
 CMDREG
 A
 A,PC
*********************
 #STKTOP
 A
 CURSET
 CMDCNT
 *

OUS DATA

 LOW,
CONTINUE COMMAND,
CONT FLAG.

MMAND IS NOT CONTINUE,
 COMMAND AND DO NEW COMMAND.

 A
 CMDWAI  WAIT
 #100    CONTINUE ?
 CNTABT

 $8C     CMPX #

 A
 STATUS
 ..CONT
 CONTO2

*****************
 2,S
 *        WAIT FOR COMMAND
```

```
00474A E12F 34    10        A           PSHS  X
00475A E131 8E    E12C      A           LDX   #CMDWAB
00476A E134 9F    00        A           STX   ..IRQE
00477A E136 35    10        A           PULS  X
00478A E138 1C    EF        A           CLI         IRQ ENABLE
00479A E13A B6    D380      A CMDWA5    LDA   ERRREG
00480A E13D 84    7F        A           ANDA  #$7F
00481A E13F B7    D380      A           STA   ERRREG
00482A E142                             BSYOFF
     A E142 7D    D40A      A           TST   BUSY
00483A E145 B6    D382      A CMDWA6    LDA   CMDREG
00484A E148 26    07   E151             BNE   CMDWA9
00485A E14A B6    D380      A           LDA   ERRREG
00486A E14D 2B    EB   E13A             BMI   CMDWA5
00487A E14F 20    F4   E145             BRA   CMDWA6
00488A E151                   CMDWA9    BSYON
     A E151 B7    D40A      A           STA   BUSY
00489A E154 1A    10                    SEI         IRQ DISABLE
00490A E156 0F    00        A           CLR   ..IRQE
00491A E158 0F    01        A           CLR   ..IRQE+1
00492A E15A 39                          RTS
00493                                   *
00494                                   **********************************
00495                                   TTL   INIT - INITIALIZE CODE -
00497                                   *
00498                                   *  INIT  COMMAND
00499                                   *
00500                                   *     .COMAR+3  BACK COLOR CODE
00501                                   *     .COMAR+4  NO. OF COLUMN ON LINE
00502                                   *     .COMAR+5  NO. OF LINE ON DISPLAY
00503                                   *     .COMAR+6  START LINE NO. OF SCROLL
00504                                   *     .COMAR+7  NO. OF SCROLL LINE
00505                                   *     .COMAR+8  INDICATER FLAG
00506                                   *
00507                  E15B         A INIT      EQU   *
00508A E15B 0F    20        A           CLR   PUTCFG
00509A E15D BD    ED9F      A           JSR   CURCLR
00510A E160 CE    D380      A           LDU   #.COMAR
00511A E163 A6    43        A           LDA   3,U
00512A E165 97    2D        A           STA   ..BCL     SET BACK COLOR
00513A E167 5F                          CLRB
00514A E168 8E    0001      A           LDX   #1
00515A E16B A6    44        A           LDA   4,U
00516A E16D 81    50        A           CMPA  #CSCX1
00517A E16F 27    08   E179             BEQ   INIT21
00518A E171 81    28        A           CMPA  #CSCX2
00519A E173 26    5E   E1D3             BNE   INIT90    WIDTH ERROR
00520A E175 C6    08        A           LDB   #8        CHAR WIDTH FLAG=8=LARGE
00521A E177 30    01        A           LEAX  1,X
00522A E179 97    34        A INIT21    STA   ..LNC
00523A E17B D7    2F        A           STB   ..CWD     CHAR WIDTH FLAG SET
00524A E17D 9F    37        A           STX   ..VCU     UNIT OF STEP
00525A E17F 5F                          CLRB
00526A E180 8E    0280      A           LDX   #GXBSZ*CFH
00527A E183 A6    45        A           LDA   5,U       SET NO. OF LINE
00528A E185 81    19        A           CMPA  #CSCY1
00529A E187 27    09   E192             BEQ   INIT31
00530A E189 81    14        A           CMPA  #CSCY2
00531A E18B 26    46   E1D3             BNE   INIT90    HEIGHT ERROR
00532A E18D C6    02        A           LDB   #2        CHAR HEIGHT FLAG=2=UNDER MARGIN
00533A E18F 8E    0320      A           LDX   #GXBSZ*(CFH+2)
00534A E192 97    35        A INIT31    STA   ..DNL
00535A E194 D7    30        A           STB   ..CHT     CHAR HEIGHT FLAG SET
00536A E196 9F    39        A           STX   ..VLU     UNIT OF LINE STEP
00537A E198 9F    8B        A           STX   HKFLG
00538A E19A E6    48        A           LDB   8,U
00539A E19C D7    36        A           STB   ..FS      INDICATE FLAG
00540A E19E 27    02   E1A2             BEQ   INIT40
00541A E1A0 4A                          DECA
00542A E1A1 4A                          DECA            USE 2 LINE FOR INDICATE
00543A E1A2 97    41        A INIT40    STA   ..DMAP+2*3 INDICATE START LINE
00544A E1A4 97    40        A           STA   ..DMAP+2*2+1 PAGE2 END LINE
00545A E1A6 A1    46        A           CMPA  6,U
00546A E1A8 23    29   E1D3             BLS   INIT90    SCROLL START ERROR
00547A E1AA A6    46        A           LDA   6,U
00548A E1AC 97    3B        A           STA   ..DMAP+2*0 SCROLL START LINE
00549A E1AE 97    3E        A           STA   ..DMAP+2*1+1 PAGE1 END LINE
00550A E1B0 AB    47        A           ADDA  7,U
00551A E1B2 91    41        A           CMPA  ..DMAP+2*3
00552A E1B4 22    1D   E1D3             BHI   INIT90
00553A E1B6 97    3C        A           STA   ..DMAP+2*0+1 SCROLL END
00554A E1B8 97    3F        A           STA   ..DMAP+2*2 PAGE2 START
00555A E1BA 0F    3D        A           CLR   ..DMAP+2*1 PAGE1 START 0,CONSTANT
00556A E1BC 96    35        A           LDA   ..DNL
00557A E1BE 97    42        A           STA   ..DMAP+2*3+1 INDICATE END,DISPLAY END
00558A E1C0 6D    47        A           TST   7,U
00559A E1C2 2B    0F   E1D3             BMI   INIT90    SCROLL NEG.
00560A E1C4 A6    4A        A           LDA   10,U      SET CONSOLE COLOR DISABLE SWITCH
00561A E1C6 27    02   E1CA             BEQ   INIT50
00562A E1C8 86    FF        A           LDA   #$FF
00563A E1CA 97    21        A INIT50    STA   CSLCDS
00564A E1CC A6    49        A           LDA   9,U
00565A E1CE 27    17   E1E7             BEQ   HOME      NOT ERASE
00566A E1D0 7E    E263      A           JMP   ERSALL    ERASE
00567                                   *
00568A E1D3 86    3C        A INIT90    LDA   #60       ERROR CODE
00569A E1D5 B7    D380      A           STA   ERRREG
00570A E1D8 8E    E0C0      A DEFALT    LDX   #INPARM
00571A E1DB 108E  D383      A           LDY   #CMDPRM
00572A E1DF C6    08        A           LDB   #8
00573A E1E1 BD    E44E      A           JSR   MOVE
00574A E1E4 7E    E15B      A           JMP   INIT      DEFAULT SET
00575                                   *
00576                                   **********************************************
00577                                   *
00578                  E1E7         A HOME      EQU   *
00579A E1E7 BD    EDBE      A           JSR   INDICT
```

```
00580A E1EA CE   D03B    A          LDU    #..DMAP
00581A E1ED 5F                      CLRB
00582A E1EE 5C            HOME10 INCB
00583A E1EF A6   C0      A          LDA    ,U+
00584A E1F1 A1   C0      A          CMPA   ,U+
00585A E1F3 27   F9   E1EE          BEQ    HOME10
00586A E1F5 97   4B      A          STA    .MDRG
00587A E1F7 97   43      A          STA    .BPY
00588A E1F9 D7   4D      A          STB    .MDRG+2
00589A E1FB E6   5F      A          LDB    -1,U
00590A E1FD D7   4C      A          STB    .MDRG+1
00591A E1FF 0F   44      A BFPTRL CLR    .BPX
00592A E201 8D   02   E205 BUFPTR BSR    BBADR
00593A E203 20   1C   E221          BRA    BVADR
00594                     *
00595            E205    A BBADR  EQU    *
00596A E205 8D   08   E20F          BSR    BADRN
00597A E207 9F   45      A          STX    .BPAD
00598A E209 109F 47      A          STY    .BAAD
00599A E20C 39                      RTS
00600                     *
00601A E20D 5F              BADRL  CLRB
00602A E20E                         SKIP2
     A E20E      8C      A          FCB    $8C     CMPX #
00603A E20F DC   43      A BADRN  LDD    .BPY    CURRENT BUFFER POINTER
00604            E211    A BADR   EQU    *
00605A E211 34   04      A          PSHS   B
00606A E213 D6   34      A          LDB    ..LNC
00607A E215 3D                      MUL
00608A E216 EB   E0      A          ADDB   ,S+
00609A E218 89   C0      A          ADCA   #..CRB/256
00610A E21A 1F   01      A          TFR    D,X
00611A E21C 31   89 07D0 A          LEAY   ..ATB-..CRB,X
00612A E220 39                      RTS
00613                     *
00614            E221    A BVADR  EQU    *
00615A E221 8D   05   E228          BSR    BVADN
00616A E223 DD   49      A          STD    .BVAD
00617A E225 39                      RTS
00618                     *
00619A E226 5F              BVADL  CLRB
00620A E227                         SKIP2
     A E227      8C      A          FCB    $8C     CMPX #
00621A E228 DC   43      A BVADN  LDD    .BPY    CURRENT BUF PTR
00622            E22A    A BVAD   EQU    *
00623A E22A 34   02      A          PSHS   A
00624A E22C 96   38      A          LDA    ..VCU+1
00625A E22E 3D                      MUL
00626A E22F 34   06      A          PSHS   D
00627A E231 A6   62      A          LDA    2,S     Y
00628A E233 D6   3A      A          LDB    ..VLU+1 LSB
00629A E235 3D                      MUL
00630A E236 E3   E4      A          ADDD   ,S
00631A E238 ED   E4      A          STD    ,S
00632A E23A A6   62      A          LDA    2,S
00633A E23C D6   39      A          LDB    ..VLU   MSB
00634A E23E 3D                      MUL
00635A E23F 1E   89      A          EXG    A,B
00636A E241 E3   E4      A          ADDD   ,S
00637A E243 32   63      A          LEAS   3,S
00638A E245 39                      RTS
00640                     *
00641                     *  ERASE  COMMAND
00642                     *
00643                     *    .COMAR+3  WINDOW NO.
00644                     *    .COMAR+4 BACK COLOR CODE
00645                     *    .COMAR+5 COLOR CODE (NEW ERASE COMMAND)
00646                     *
00647A E246 0F   20      A NERASE CLR    PUTCFG
00648A E248 BD   ED9F    A          JSR    CURCLR
00649A E24B F6   D385    A          LDB    .COMAR+5 LOAD COLOR.
00650A E24E 20   07   E257          BRA    ERASDO
00651                     *
00652            E250    A ERASE  EQU    *
00653A E250 0F   20      A          CLR    PUTCFG
00654A E252 BD   ED9F    A          JSR    CURCLR
00655A E255 D6   2C      A          LDB    ..ACL   LOAD THE CURRENT COLOR CODE
00656            E257    A ERASDO EQU    *
00657A E257 B6   D383    A          LDA    .COMAR+3
00658A E25A 26   41   E29D          BNE    ERASE0
00659A E25C D7   2C      A          STB    ..ACL   SET THE COLOR CODE.
00660A E25E B6   D384    A          LDA    .COMAR+4
00661A E261 97   2D      A          STA    ..BCL   SET BACK COLOR
00662            E263    A ERSALL EQU    *
00663A E263 8E   0000    A          LDX    #0
00664A E266 CE   4000    A          LDU    #$4000
00665A E269 00   21      A          NEG    CSLCDS
00666A E26B                         SCROFF
     A E26B B7   D408    A          STA    CRTON
00667A E26E BD   E2F8    A          JSR    ERSS04
00668A E271                         SCRON
     A E271 B5   D408    A          BITA   CRTON
00669A E274 00   21      A          NEG    CSLCDS
00670A E276 96   3D      A          LDA    ..DMAP+2 TOP VRAM Y
00671A E278 D6   42      A          LDB    ..DMAP+2*4-1 BOTTOM VRAM Y PAST 1
00672A E27A DD   73      A          STD    .CVTOP
00673A E27C BD   E335    A          JSR    ERSBUF
00674A E27F BD   E1E7    A          JSR    HOME
00675A E282 CE   D03B    A ASSFLD LDU    #..DMAP ASSUME FIELD MARK
00676A E285 A6   C1      A ERSAL5 LDA    ,U++
00677A E287 BD   E20D    A          JSR    BADRL
00678A E28A A6   A4      A          LDA    ,Y
00679A E28C 8A   80      A          ORA    #AMFL
00680A E28E A7   A4      A          STA    ,Y
00681A E290 1183 D043    A          CMPU   #..DMAP+8
00682A E294 22   06   E29C          BHI    ERSAL9
00683A E296 25   ED   E285          BLO    ERSAL5
```

```
00684A E298 33   5F      A          LEAU
00685A E29A 20   E9   E285          BRA
00686A E29C 39              ERSAL9 RTS
00687                     *
00688A E29D 81   03      A ERASE0 CMPA
00689A E29F 1022 FE25 E0C8          LBHI
00690A E2A3 D7   2C      A          STB
00691            E2A5    A ERASE1 EQU
00692A E2A5 48                      LSLA
00693A E2A6 8E   D039    A          LDX
00694A E2A9 30   86      A          LEAX
00695A E2AB E6   84      A          LDB
00696A E2AD E1   01      A          CMPB
00697A E2AF 1027 FE15 E0C8          LBEQ
00698A E2B3 44                      LSRA
00699A E2B4 97   4D      A          STA
00700A E2B6 EC   84      A          LDD
00701A E2B8 4D                      TSTA
00702A E2B9 26   04   E2BF          BNE
00703A E2BB D1   41      A          CMPB
00704A E2BD 27   A4   E263          BEQ
00705            E2BF    A ERASES EQU
00706A E2BF DD   73      A          STD
00707A E2C1 DD   4B      A          STD
00708A E2C3 97   43      A          STA
00709A E2C5 0F   44      A          CLR
00710A E2C7 8D   60   E329          BSR
00711A E2C9 96   4B      A ASMFLD LDA
00712A E2CB BD   E20D    A          JSR
00713A E2CE A6   A4      A          LDA
00714A E2D0 8A   80      A          ORA
00715A E2D2 A7   A4      A          STA
00716A E2D4 7E   E201    A          JMP
00718                     *
00719                     *
00720                     *    ERASE DIS
00721                     *    AND ATTRI
00722                     *
00723                     *
00724                     * CVTOP   STAR
00725                     * CVTOP+1 PAST
00726                     *
00727            E2D7    A ERSVRM EQU
00728A E2D7 96   74      A          LDA
00729A E2D9 BD   E226    A          JSR
00730A E2DC 1F   03      A          TFR
00731A E2DE 83   4000    A          SUBD
00732A E2E1 23   0E   E2F1          BLS
00733A E2E3 CE   4000    A          LDU
00734A E2E6 34   06      A          PSHS
00735A E2E8 8D   07   E2F1          BSR
00736A E2EA 8E   0000    A          LDX
00737A E2ED 35   40      A          PULS
00738A E2EF 20   07   E2F8          BRA
00739            E2F1    A ERSS03 EQU
00740A E2F1 96   73      A          LDA
00741A E2F3 BD   E226    A          JSR
00742A E2F6 1F   01      A          TFR
00743A E2F8 96   21      A ERSS04 LDA
00744A E2FA 27   1F   E31B          BEQ
00745A E2FC 2B   0B   E309          BMI
00746A E2FE 86   04      A          LDA
00747A E300 D6   2C      A          LDB
00748A E302 C5   08      A          BITB
00749A E304 26   1D   E323          BNE
00750A E306 4F                      CLRA
00751A E307 20   1A   E323          BRA
00752A E309 30   89 8000 A ERSMNO LEAX
00753A E30D 33   C9 8000 A          LEAU
00754A E311 4F                      CLRA
00755A E312 D6   2C      A          LDB
00756A E314 C5   08      A          BITB
00757A E316 27   54   E36C          BEQ
00758A E318 4C                      INCA
00759A E319 20   51   E36C          BRA
00760A E31B 96   2C      A ERSVN  LDA
00761A E31D 85   08      A          BITA
00762A E31F 26   02   E323          BNE
00763A E321 96   2D      A          LDA
00764            E323    A ERSS05 EQU
00765A E323 8D   47   E36C          BSR
00766A E325 8D   45   E36C          BSR
00767A E327 20   43   E36C          BRA
00768                     *
00769            E329    A ERSSOF EQU
00770A E329                         SCROFF
     A E329 B7   D408    A          STA
00771A E32C 00   21      A ERSSUB NEG
00772A E32E 8D   A7   E2D7          BSR
00773A E330 00   21      A          NEG
00774A E332                         SCRON
     A E332 B5   D408    A          BITA
00775A E335 96   74      A ERSBUF LDA
00776A E337 BD   E20D    A          JSR
00777A E33A 34   20      A          PSHS
00778A E33C 33   84      A          LEAU
00779A E33E 96   73      A          LDA
00780A E340 BD   E20D    A          JSR
00781A E343 34   20      A          PSHS
00782A E345 4F                      CLRA
00783A E346 8D   04   E34C          BSR
00784A E348 96   2C      A          LDA
00785A E34A 35   50      A          PULS
00786                     *
00788            E34C    A ERSARA EQU
00789                     *
00790                     * A STORED DAT
```

```
-1,U
ERSAL5


#3
ERR60
..ACL    SET THE COLOR CODE.
*

#..DMAP-2
A,X
O,X
1,X
ERR60

.MDRG+2 KIND
,X
         IS THE CURRENT WINDOW
ERASE5   FULL OF LOGICAL DISPLAY AREA?
..DMAP+2*3
ERSALL   YES.
*
.CVTOP
.MDRG
.BPY
.BPX
ERSSOF
.MDRG    ASSUME FIELD TAG AT CURRENT WINDOW TOP.
BADRL
,Y
#AMFL
,Y
BUFPTR


PLAY,CHARACTER BUFFER
BUTE BUFFER


T LINE OF ERASE
 END LINE OF ERASE


.CVTOP+1 GET ADDRESS THAT IS PAST 1
BVADL    TO END OF ERASE AREA.
D,U      SET U.
#$4000   OUT OF BLOCK?
ERSS03   NO.
#$4000
D
ERSS03
#0
U
ERSS04
*
.CVTOP   GET TOP ADDRESS OF ERASE.
BVADL
D,X      SET X.
CSLCDS   COLOR ENABLE?
ERSVN    YES.
ERSMNO   NO,WILL I ERASE THE ONLY GREEN?
#4       NO.
..ACL
#AMRV    REVERSE?
ERSS05   YES ,DISPLAY IS ERASED BY GREEN.

ERSS05   DISPLAY IS ERASED BY BLACK.
..VG,X   I WILL ERASE ONLY GREEN VRAM.
..VG,U

..ACL
#AMRV
ERSV

ERSV
..ACL    IS IT REVERCE?
#AMRV
ERSS05   YES, ERASE COLOR IS CHAR'S COLOR.
..BCL    NO, ERASE COLOR IS BACK COLOR.
*
ERSV     BLUE
ERSV     RED
ERSV     GREEN & RETURN


*

CRTON
CSLCDS
ERSVRM
CSLCDS

CRTON
.CVTOP+1 GET BUFFER ADDR THAT IS
BADRL    PAST 1 TO END OF ERASE AREA.
Y        SAVE ATTR ADDR.
,X       SET U CHAR ADDR
.CVTOP   GET BUFFER ADDRESS THAT
BADRL    IS TOP OF ERASE AREA.
Y        SAVE ATTR ADDRR.

ERSARA   ERASE CHAR BUFFER.
..ACL
X,U
           ERASE ATTR BUFFER.

*

A
```

```
                                  * X START ADDR
                                  * U END ADDR
                                  *
                                  * THESE PARAMETERS MUST BE AS FOLLOWS.
                                  *
                                  * THE X IS LESS THAN THE U.
                                  * SIZE (U-X) IS EVEN NUMBER AND
                                  * GREATER THAN 6.
                                  *
A E34C 31    02        A          LEAY    2,X
A E34E 30    06        A          LEAX    6,X
A E350 34    30        A          PSHS    X,Y
A E352 1F    89        A          TFR     A,B       MAKE DATA
A E354 1F    01        A          TFR     D,X
A E356 31    84        A          LEAY    ,X
A E358 36    36        A ERSAR3   PSHU    D,X,Y     SET 6 BYTES.
A E35A 11A3  E4        A          CMPU    ,S        IS THE REMAINDER .GE. 6?
A E35D 24    F9   E358            BHS     ERSAR3    YES.
A E35F 11A3  62        A          CMPU    2,S       HOW MANY BYTES IS REMAIND?
A E362 25    06   E36A            BLO     ERSAR9    IT IS 0.
A E364 27    02   E368            BEQ     ERSAR8    IT IS 2.
A E366 ED    C3        A          STD     ,--U      IT IS 4.
A E368 ED    C3        A ERSAR8   STD     ,--U
A E36A                   ERSAR9   RTSPUL  (D,Y)
A E36A 35    A6        A          PULS    D,Y,PC
                                  *
A E36C 44                ERSV     LSRA
A E36D 34    52        A          PSHS    A,X,U
A E36F 86    00        A          LDA     #$00
A E371 24    01   E374            BCC     ERSV05
A E373 43                         COMA
A E374                   ERSV05   VACON
  A E374 B5  D409      A          BITA    VACC
A E377 8D    D3   E34C            BSR     ERSARA
A E379                            VACOFF
  A E379 B7  D409      A          STA     VACC
A E37C 35    52        A          PULS    A,X,U
A E37E 30    89 4000 A            LEAX    $4000,X TOP ADDRESS OF NEXT COLOR.
A E382 33    C9 4000 A            LEAU    $4000,U END ADDRESS OF NEXT COLOR.
A E386 39                         RTS
                                  TTL     COMMON SUBROUTINE
                                  * GAXC :TEST GRAPHIC CO-ORDINATE X,Y
                                  *
                    E387 A GAXC   EQU     *
A E387 9E    5A        A          LDX     .GX
A E389 109E  5B        A          LDY     .GY
                    E38C A GAXCR  EQU     *
A E38C 8C    0280      A GAXC1    CMPX    #GXD
A E38F 24    04   E395            BHS     GAXC2
A E391 108C  00C8      A          CMPY    #GYD
A E395 39                GAXC2    RTS
                                  *
                                  *
A E396       80        A BAS      FCB     $80
A E397       40        A          FCB     $40
A E398       20        A          FCB     $20
A E399       10        A          FCB     $10
A E39A       08        A          FCB     $08
A E39B       04        A          FCB     $04
A E39C       02        A          FCB     $02
A E39D       01        A          FCB     $01
                                  *
                                  *
                                  *
                                  *
                                  *
                                  * GPGETR : READ COLOR CODE
                                  **
                    E39E A GPGETR EQU     *
A E39E                   GPGET1   VACON
  A E39E B5  D409      A          BITA    VACC
A E3A1 4F                         CLRA
A E3A2 E5    C4        A          BITB    ..VB,U
A E3A4 27    01   E3A7            BEQ     GPGET2
A E3A6 4C                         INCA
A E3A7 E5    C9 4000 A GPGET2     BITB    ..VR,U
A E3AB 27    02   E3AF            BEQ     GPGET3
A E3AD 8B    02        A          ADDA    #2
A E3AF E5    C9 8000 A GPGET3     BITB    ..VG,U
A E3B3 27    02   E3B7            BEQ     GPGET4
A E3B5 8B    04        A          ADDA    #4
A E3B7                   GPGET4   VACOFF
  A E3B7 B7  D409      A          STA     VACC
A E3BA 39                         RTS
                                  *
                                  *
                                  *
                                  *
                                  * CFADR : MAKE CHARACTER FONT ADDRESS
                                  *
                    E3BB A CFADR  EQU     *
A E3BB 34    06        A          PSHS    A,B
A E3BD C6    08        A          LDB     #CFH
A E3BF 3D                         MUL
A E3C0 C3    D800      A          ADDD    #..CFNT
A E3C3 1F    01        A          TFR     D,X
A E3C5                            RTSPUL  (A,B)
  A E3C5 35  86        A          PULS    A,B,PC
                                  *
                                  *
                                  *
                                  *
                                  * CBADR : CALCULATE BUFFER ADDRESS
                                  *
                    E3C7 A CBADR  EQU     *
A E3C7 DC    4E        A          LDD     .CY
A E3C9 BD    E211      A          JSR     BADR
```

```
00894A E3CC 9F   50       A        STX    .CBAD
00895A E3CE 109F 52       A        STY    .CAAD
00896A E3D1 DC   4E       A        LDD    .CY
00897A E3D3 BD   E22A     A        JSR    BVAD
00898A E3D6 DD   54       A        STD    .CVAD
00899A E3D8 39                     RTS
00900                     *
00901                     *
00902                     *
00903                     *
00904                     *
00905                     * CAXCR : TEST CHARACTER CO-ORDINATE
00906                     *
00907          E3D9       A CAXCR  EQU    *
00908A E3D9 91   34       A        CMPA   ..LNC
00909A E3DB 24   02  E3DF          BCC    CAXC2
00910A E3DD D1   41       A        CMPB   ..DMAP+2*3
00911A E3DF 39              CAXC2  RTS
00912                     *
00913                     *
00914          E3E0       A GADDR  EQU    *
00915A E3E0 34   32       A        PSHS   A,X,Y
00916A E3E2 EC   61       A        LDD    1,S
00917A E3E4 8D   12  E3F8          BSR    GXOFFR
00918A E3E6 1E   02       A        EXG    D,Y
00919A E3E8 86   50       A        LDA    #GXBSZ
00920A E3EA 3D                     MUL
00921A E3EB 33   8B       A        LEAU   D,X
00922A E3ED CC   E396     A        LDD    #BAS
00923A E3F0 1E   02       A        EXG    D,Y
00924A E3F2 E6   A5       A        LDB    B,Y
00925A E3F4                        RTSPUL (A,X,Y)
     A E3F4 35   B2       A        PULS   A,X,Y,PC
00926                     *
00927                     *
00928                     *  GOXFF: CALCULATE OFFSET VALUE AND BIT NUMBER
00929                     *              OF GRAPHIC X CO-ORDINATE
00930                     *
00931                     *
00932          E3F6       A GXOFF  EQU    *
00933A E3F6 1F   01       A        TFR    D,X     TRASFER X CO-ORDINATE TO X
00934A E3F8 8D   05  E3FF GXOFFR BSR    LSRD3
00935A E3FA 1E   01       A        EXG    D,X     SET OFFSET VALUE
00936A E3FC C4   07       A        ANDB   #$07    SET BIT NUMBER
00937A E3FE 39                     RTS
00938                     *
00939                     * END GXOFF
00940                     *
00941          E3FF       A LSRD3  EQU    *
00942A E3FF 44                     LSRA
00943A E400 56                     RORB
00944A E401 44                     LSRA
00945A E402 56                     RORB
00946A E403 44                     LSRA
00947A E404 56                     RORB
00948A E405 39                     RTS
00949                     *
00950A E406 8E   D383     A BOXINV LDX    #CMDPRM
00951A E409 108E D05F     A        LDY    #.BOXX1
00952A E40D C6   08       A        LDB    #8
00953A E40F 8D   3D  E44E          BSR    MOVE
00954A E411 8D   0B  E41E          BSR    BOX
00955A E413 DC   63       A        LDD    .BOXX2
00956A E415 93   5F       A        SUBD   .BOXX1
00957A E417 1F   01       A        TFR    D,X
00958A E419 DC   65       A        LDD    .BOXY2
00959A E41B 93   61       A        SUBD   .BOXY1
00960A E41D 39                     RTS
00961                     *
00962                     *
00963                     *  BOX: TEST TWO POINT CO-ORDINATE
00964                     *
00965                     *
00966          E41E       A BOX    EQU    *
00967A E41E CE   D05F     A        LDU    #.BOXX1 TOP ADDRESS U
00968A E421 AE   C4       A BOXR   LDX    ,U
00969A E423 10AE 42       A        LDY    2,U
00970A E426 8D   1E  E446          BSR    GAXCHK
00971A E428 AE   44       A        LDX    4,U
00972A E42A 10AE 46       A        LDY    6,U
00973A E42D 8D   17  E446          BSR    GAXCHK
00974A E42F AC   C4       A        CMPX   ,U
00975A E431 24   06  E439          BHS    BOX1
00976A E433 EC   C4       A        LDD    ,U
00977A E435 AF   C4       A        STX    ,U
00978A E437 ED   44       A        STD    4,U
00979A E439 10AC 42       A BOX1   CMPY   2,U
00980A E43C 24   07  E445          BHS    BOX2
00981A E43E EC   42       A        LDD    2,U
00982A E440 10AF 42       A        STY    2,U
00983A E443 ED   46       A        STD    6,U
00984A E445 39              BOX2   RTS
00985                     *
00986A E446 BD   E38C     A GAXCHK JSR    GAXCR   COORDINATES CHECK
00987A E449 25   FA  E445          BCS    BOX2    OK.
00988A E44B 7E   E0D1     A        JMP    ERR63   ERROR
00989                     *
00990                     * END BOX
00991                     **
00992                     *
00993                     *
00994                     *
00995                     * MOVE: MOVE DATA
00996                     *
00997          E44E       A MOVE   EQU    *
00998A E44E A6   80       A        LDA    ,X+     LOAD MOVE DATA
00999A E450 A7   A0       A        STA    ,Y+     STORE MOVE DATA
```

```
01000A E452 5A                     DECB
01001A E453 26   F9  E44E          BNE    MOVE
01002A E455 39                     RTS
01003                     *
01004                     * END MOVE
01005                     *
01006                     *
01008                     * SET UP LINEX ROUTINE
01009                     *
01010A E456 7F   D096     A INITLN CLR    LNCHR
01011A E459 DC   56       A        LDD    .GCL    LOAD
01012A E45B 8E   F060     A        LDX    #LNFUNC LOAD
01013A E45E CE   D082     A        LDU    #BLULIN LOAD
01014A E461 8D   12  E475          BSR    LENTRY  SET
01015A E463 DC   56       A        LDD    .GCL    SEP.
01016A E465 8E   F02E     A        LDX    #LNXFNC PARR
01017A E468 CE   D06F     A        LDU    #BLUSID
01018A E46B 8D   08  E475          BSR    LENTRY
01019                     *               AND D
01020                     *
01021                     * SET UP GPDSP ROUTINE
01022                     * PICKUP AND SET VECTOR TO
01023                     *
01024          E46D       A GPDSPI EQU    *
01025A E46D DC   56       A        LDD    .GCL    A=CO
01026A E46F 8E   E4B1     A GPDSIR LDX    #GPDFNC MAKE
01027A E472 CE   D079     A FENTRY LDU    #BLUDSP LOAD
01028A E475 C1   01       A LENTRY CMPB   #1      ID T
01029A E477 26   02  E47B          BNE    NOTPRE  NO.
01030A E479 96   2D       A        LDA    ..BCL   YES,
01031A E47B 58              NOTPRE LSLB
01032A E47C 58                     LSLB           MAKE
01033A E47D 3A                     ABX            FUNC
01034A E47E C6   7E       A        LDB    #$7E    LOAD
01035A E480 8D   02  E484          BSR    SETENT  SET
01036A E482 8D   00  E484          BSR    SETENT  SET
01037                     *                SET
01038A E484 E7   C0       A SETENT STB    ,U+     SET
01039A E486 44                     LSRA           SET
01040A E487 34   06       A        PSHS   D       SAVE
01041A E489 EC   84       A        LDD    ,X      ASSU
01042A E48B 25   02  E48F          BCS    CBITON  IS I
01043A E48D EC   02       A        LDD    2,X     NO,
01044A E48F ED   C1       A CBITON STD    ,U++    SET
01045A E491                        RTSPUL D
     A E491 35   86       A        PULS   D,PC
01046                     *
01047                     *
01048                     * GPDSP : DISPLAY DOT ON G
01049                     *
01050                     *
01051          E493       A GPDSP  EQU    *
01052A E493 9E   5A       A        LDX    .GX     LOAD
01053A E495 109E 58       A        LDY    .GY     LOAD
01054A E498 BD   E3E0     A GPDSPC JSR    GADDR   VRAM
01055A E49B 34   04       A GPDSPR PSHS   B       SAVE
01056A E49D 9D   79       A        JSR    BLUDSP
01057A E49F E6   E4       A        LDB    ,S
01058A E4A1 33   C9 4000  A        LEAU   ..VR,U
01059A E4A5 9D   7C       A        JSR    REDDSP
01060A E4A7 E6   E4       A        LDB    ,S
01061A E4A9 33   C9 4000  A        LEAU   ..VG-..VR,U
01062A E4AD 9D   7F       A        JSR    GREDSP
01063A E4AF                        RTSPUL B
     A E4AF 35   84       A        PULS   B,PC
01064                     *
01065A E4B1      E4C5     A GPDFNC FDB    GPORA,GPAND
     A E4B3      E4D1     A
01066A E4B5      E4C5     A        FDB    GPORA,GPAND
     A E4B7      E4D1     A
01067A E4B9      E4C5     A        FDB    GPORA,GPNOP
     A E4BB      E4D0     A
01068A E4BD      E4D0     A        FDB    GPNOP,GPAND
     A E4BF      E4D1     A
01069A E4C1      E4DE     A        FDB    GPEOR,GPNOP
     A E4C3      E4D0     A
01070                     *
01071                     *
01072          E4C5       A GPORA  EQU    *
01073A E4C5                        VACON
     A E4C5 B5   D409     A        BITA   VACC
01074A E4C8 12                     NOP
01075A E4C9 EA   C4       A        ORB    0,U
01076A E4CB E7   C4       A        STB    0,U
01077A E4CD                        VACOFF
     A E4CD B7   D409     A        STA    VACC
01078A E4D0 39              GPNOP  RTS
01079                     *
01080                     *
01081          E4D1       A GPAND  EQU    *
01082A E4D1 53                     COMB
01083A E4D2                        VACON
     A E4D2 B5   D409     A        BITA   VACC
01084A E4D5 12                     NOP
01085A E4D6 E4   C4       A        ANDB   0,U
01086A E4D8 E7   C4       A        STB    0,U
01087A E4DA                        VACOFF
     A E4DA B7   D409     A        STA    VACC
01088A E4DD 39                     RTS
01089                     **
01090                     *
01091          E4DE       A GPEOR  EQU    *
01092A E4DE                        VACON
     A E4DE B5   D409     A        BITA   VACC
01093A E4E1 12                     NOP
01094A E4E2 E8   C4       A        EORB   0,U
01095A E4E4 E7   C4       A        STB    0,U
```

```
 COLOR AND FUNCTION CODE.
 FUNCTION ENTRY TABLE ADDRESS.
 RAM HOOK TABLE ADDRESS.
THE RAM HOOKS.
 3 1981
ALLEL TO X AXIS LINE'S SIDE TO SET ENTRY

O GPDSP INITIALIZE.

 DISPLAY

LOR,B=FUNCTION
 RAM HOOK FOR DOT DISPLAY.
 RAM HOOK ADDRESS.
HE FUNCTION CODE PRESET?

 USE BACK COLOR.

 ADDRESS OF CURRENT
TION'S ENTRY TABLE.
 JUMP OPERATION CODE.
BLUE'S RAM HOOK
RED'S RAM HOOK
 GREEN'S RAM HOOK.
JUMP.
COLOR BIT TO C.

ME BIT-ON'S ENTRY
T ON? YES.
LOAD BIT-OFF'S ENTRY.
ENTRRY ADDRESS.

RAPHIC CO-ORDINATE

 X COORDINATES
 Y COORDINATES
 ADDR. MADE
 BIT MASK

SET
PRESET
OR
AND
EOR
```

```
01096A E4E6                         VACOFF
    A E4E6 B7   D409    A           STA    VACC
01097A E4E9 39                      RTS
01098                       *
01099                       *  END GPDSP
01100                       *
01102                       *
01103                       * 'CDSP' DISPLAY CHARACTER
01104                       *
01105                       * A IS THE CHARACTER CODE.
01106                       * B IS THE COLOR CODE.
01107                       * X IS THE DISPLAY ADDRESS.
01108                       *
01109A E4EA 34   16      A CDSINT PSHS   D,X
01110A E4EC DD   8B      A        STD    HKFLG
01111A E4EE 8E   E5D4    A        LDX    #CDJMP
01112A E4F1 D6   2F      A        LDB    ..CWD
01113A E4F3 3A                    ABX
01114A E4F4 108E D08D    A        LDY    #HKBLUE
01115A E4F8 86   03      A        LDA    #3
01116A E4FA C6   7E      A CDSIN5 LDB    #$7E
01117A E4FC E7   A0      A        STB    ,Y+      JUMP CODE
01118A E4FE 5F                    CLRB
01119A E4FF 64   E4      A        LSR    ,S
01120A E501 59                    ROLB
01121A E502 64   61      A        LSR    1,S
01122A E504 59                    ROLB
01123A E505 58                    LSLB
01124A E506 EE   85      A        LDU    B,X
01125A E508 EF   A1      A        STU    ,Y++
01126A E50A 4A                    DECA
01127A E50B 26   ED   E4FA        BNE    CDSIN5
01128A E50D                       RTSPUL (D,X)
    A E50D 35   96      A         PULS   D,X,PC
01129                       *
01130            E50F    A BITEXT EQU    *        EXTEND A'S LOWER 4
01131A E50F 44                    LSRA            BITS TO 8BITS ON B.
01132A E510 56                    RORB            RETURNS A IS LOGICAL
01133A E511 57                    ASRB            SHIFTED RIGHT 4.
01134A E512 44                    LSRA
01135A E513 56                    RORB
01136A E514 57                    ASRB
01137A E515 44                    LSRA
01138A E516 56                    RORB
01139A E517 57                    ASRB
01140A E518 44                    LSRA
01141A E519 56                    RORB
01142A E51A 57                    ASRB
01143A E51B 39                    RTS
01144                       *
01145A E51C 9E   54      A CDSPC  LDX    .CVAD
01146A E51E                       SKIP2
    A E51E      8C      A         FCB    $8C      CMPX #
01147A E51F 9E   49      A CDSPB  LDX    .BVAD
01148            E521    A CDSP   EQU    *
01149A E521 33   84      A        LEAU   ,X
01150A E523 BD   E3BB    A        JSR    CFADR    X=FONT ADDR.
01151A E526 96   21      A        LDA    CSLCDS
01152A E528 27   0E   E538        BEQ    CDSPN
01153A E52A C5   08      A        BITB   #AMRV
01154A E52C 26   05   E533        BNE    CDSMRV
01155A E52E CC   0004    A        LDD    #0*256+4
01156A E531 20   0D   E540        BRA    CDSP0
01157A E533 CC   0400    A CDSMRV LDD    #4*256+0
01158A E536 20   08   E540        BRA    CDSP0
01159A E538 96   2D      A CDSPN  LDA    ..BCL    BACK COLOR LOADED.
01160A E53A C5   08      A        BITB   #AMRV    IF REVERSE IS DIRECTED
01161A E53C 27   02   E540        BEQ    CDSP0    BACK COLOR AND CHAR
01162A E53E 1E   89      A        EXG    A,B      COLOR ARE EXCHANGED.
01163A E540 34   50      A CDSP0  PSHS   X,U      SAVE COLORS,FONT ADDR
01164                       *                      AND DISPLAY ADDR.
01165A E542 84   07      A        ANDA   #$07
01166A E544 C4   07      A        ANDB   #$07
01167A E546 1093 8B      A        CMPD   HKFLG    IS REQUESTED COLOR SAME AS CURRENT COLOR?
01168A E549 27   02   E54D        BEQ    CDSP1    YES.
01169A E54B 8D   9D   E4EA        BSR    CDSINT   HK COLOR
01170            E54D    A CDSP1  EQU    *
01171A E54D 96   2F      A        LDA    ..CWD    IS THE CHARACTER WIDTH
01172                       *                      SMALL?
01173A E54F 27   19   E56A        BEQ    CDSP50   YES.
01174A E551 CE   D079    A        LDU    #EXFONT  NO, MAKE LARGE CHAR.
01175A E554 EF   E4      A        STU    ,S       FONT ADDR RESTORE.
01176A E556 C6   08      A        LDB    #CFH
01177A E558 34   04      A        PSHS   B
01178A E55A A6   80      A CDSP3  LDA    ,X+      ORIGINAL FONT LOADED.
01179A E55C 8D   B1   E50F        BSR    BITEXT   EXTEND LOWER 4.
01180A E55E E7   48      A        STB    8,U
01181A E560 8D   AD   E50F        BSR    BITEXT   EXTEND HIGHER 4BITS.
01182A E562 E7   C0      A        STB    ,U+
01183A E564 6A   E4      A        DEC    ,S       END?
01184A E566 26   F2   E55A        BNE    CDSP3    NO, GO DO MORE.
01185A E568 32   61      A        LEAS   1,S      YES, RESTORE SP..
01186A E56A 96   21      A CDSP50 LDA    CSLCDS   MONO?
01187A E56C 26   16   E584        BNE    CDSPMN   YES.
01188                       *
01189A E56E EE   E4      A        LDU    ,S
01190A E570 AE   62      A        LDX    2,S
01191A E572 CC   0008    A        LDD    #CFH
01192A E575 9D   8D      A        JSR    HKBLUE
01193A E577 EE   E4      A        LDU    ,S
01194A E579 AE   62      A        LDX    2,S
01195A E57B 30   89 4000 A        LEAX   ..VR,X
01196A E57F CC   0008    A        LDD    #CFH
01197A E582 9D   90      A        JSR    HKRED
01198A E584 EE   E4      A CDSPMN LDU    ,S
01199A E586 AE   62      A        LDX    2,S
01200A E588 30   89 8000 A        LEAX   ..VG,X
```

サブシステム・モニタ ソース・リスト

```
A E58C CC    0008    A              LDD     #CFH
A E58F 9D    93      A              JSR     HKGREE
A E591                              RTSPUL  (X,U)
  A E591 35  D0      A              PULS    X,U,PC
A E593 43                   CDSRVL  COMA            NOT REQUEST
A E594 34    14      A      CDSNML  PSHS    X,B
A E596 8D    06   E59E              BSR     CDSNMS  MSB PUT
A E598 35    14      A              PULS    X,B
A E59A 30    01      A              LEAX    1,X
A E59C                              SKIP1   *       LSB PUT
  A E59C     21      A              FCB     $21
A E59D 43                   CDSRVS  COMA            NOT REQUEST
A E59E 97    2E      A      CDSNMS  STA     ..CLEX  REQUEST BIT STRING STORE
A E5A0                              VACON
  A E5A0 B5  D409    A              BITA    VACC
A E5A3 A6    C0      A      CDSTA   LDA     ,U+
A E5A5 98    2E      A              EORA    ..CLEX
A E5A7 A7    84      A              STA     ,X
A E5A9 30    88 50   A              LEAX    GXBSZ,X
A E5AC 5A                           DECB
A E5AD 26    F4   E5A3              BNE     CDSTA
A E5AF                              VACOFF
  A E5AF B7  D409    A              STA     VACC
A E5B2 96    2E      A              LDA     ..CLEX  RESTORE A.
A E5B4                              SKIP1
  A E5B4     21      A              FCB     $21
A E5B5 43                   CDSONS  COMA
A E5B6                      CDSOFS  VACON
  A E5B6 B5  D409    A              BITA    VACC
A E5B9 DB    30      A              ADDB    ..CHT   ADD MARGIN
A E5BB 27    08   E5C5              BEQ     DSRTS   IF MARGIN IS ZERO GO RETURN.
A E5BD A7    84      A      CDCLR   STA     ,X
A E5BF 30    88 50   A              LEAX    GXBSZ,X
A E5C2 5A                           DECB
A E5C3 26    F8   E5BD              BNE     CDCLR
A E5C5                      DSRTS   VACOFF
  A E5C5 B7  D409    A              STA     VACC
A E5C8 39                           RTS
A E5C9 43                   CDSONL  COMA
A E5CA 34    14      A      CDSOFL  PSHS    X,B
A E5CC 8D    E8   E5B6              BSR     CDSOFS
A E5CE 35    14      A              PULS    X,B
A E5D0 30    01      A              LEAX    1,X
A E5D2 20    E2   E5B6              BRA     CDSOFS
                            *
A E5D4       E5B6    A      CDJMP   FDB     CDSOFS  OFF
A E5D6       E59E    A              FDB     CDSNMS  NORMAL
A E5D8       E59D    A              FDB     CDSRVS  REVERSE
A E5DA       E5B5    A              FDB     CDSONS  ON
A E5DC       E5CA    A              FDB     CDSOFL  OFF
A E5DE       E594    A              FDB     CDSNML  NORMAL
A E5E0       E593    A              FDB     CDSRVL  REVERSE
A E5E2       E5C9    A              FDB     CDSONL  ON
                                    TTL     CONSL - PUT CODE -
             D096    A      PCRORD  EQU     .WORK   CR ORDER FLAG
             D097    A      PUTGET  EQU     PCRORD+1 PUT/GET
             D098    A      PUTIX   EQU     PUTGET+1
             D09A    A      PUTFLD  EQU     PUTIX+2 ERASE FIELD FLAG
             D09B    A      PUTEND  EQU     PUTFLD+1
                            *
                            *  G E T   C O M M A N D
                            *
                            *  P U T   C O M M A N D
                            *
A E5E4                              FLGENT  PUT,GET,PUTGET
  A E5E4     86      A      GET     FCB     $86     LDA #
  A E5E5 4F                 PUT     CLRA
  A E5E6 97  97      A              STA     PUTGET
                            *
             E5E8    A      PUT000  EQU     *       DISPLAY STRING BEGIN
A E5E8 BD    E92F    A              JSR     GETRNG  DISPLAY WINDOW RANGE MADE
A E5EB 8E    D384    A              LDX     #.COMAR+4
A E5EE 9F    98      A              STX     PUTIX   STRING PTR INIT.
A E5F0 0F    96      A              CLR     PCRORD  CLEAR CR ORDER FLAG FOR SUSPEND TO PUT.
             E5F2    A      PUT010  EQU     *
A E5F2 BD    E729    A              JSR     PUTDAT  TAKE PUT CHAR
A E5F5 24    28   E61F              BCC     PUT100  EXIST
A E5F7 0D    97      A              TST     PUTGET  DATA END
A E5F9 1026  02DF E8DC              LBNE    GET000  GET OPRATERS INPUT
A E5FD 96    1F      A              LDA     .CSCTL  SUSPEND TO PUT?
A E5FF 85    10      A              BITA    #CSLSUS
A E601 27    19   E61C              BEQ     PUT099  NO.
A E603 96    96      A              LDA     PCRORD  DID WE DO CR ORDER?
A E605 27    15   E61C              BEQ     PUT099  NO.
A E607                              SEF     DISABLE KEY INTERRUPT.
  A E607 1A  40      A              ORCC    #$40
A E609 DC    03      A              LDD     KEYDAT
A E60B 4D                           TSTA
A E60C 2B    0E   E61C              BMI     PUT099
A E60E 27    0C   E61C              BEQ     PUT099
A E610 C1    1B      A              CMPB    #ESC
A E612 26    08   E61C              BNE     PUT099
A E614 0F    03      A              CLR     KEYDAT
A E616 1C    AF      A              ANDCC   #$AF    KEY AND ABEND INTERRUPT ENABLE.
A E618 96    03      A      PUT090  LDA     KEYDAT  WAIT
A E61A 27    FC   E618              BEQ     PUT090
A E61C                      PUT099  CLF     ENABLE  KEY INTERRUPT
  A E61C 1C  BF      A              ANDCC   #$BF
A E61E 39                           RTS
                            *
             E61F    A      PUT100  EQU     *
A E61F D6    1F      A              LDB     .CSCTL
A E621 C5    02      A              BITB    #CSLORD
A E623 27    1B   E640              BEQ     PUT106  ORDER DISABLE
A E625 81    20      A              CMPA    #'
A E627 24    13   E63C              BHS     PUT105
A E629 34    02      A              PSHS    A
A E62B BD    ED9F    A              JSR
A E62E 35    02      A              PULS
A E630 81    14      A              CMPA
A E632 24    12   E646              BHS
A E634 8E    E8B4    A              LDX
A E637 48                           LSLA
A E638 AD    96      A              JSR
A E63A 20    B6   E5F2              BRA
                            *
A E63C 81    7F      A      PUT105  CMPA
A E63E 27    B2   E5F2              BEQ
A E640 8D    21   E663      PUT106  BSR
A E642 0F    1C      A              CLR
A E644 20    AC   E5F2              BRA
                            *
A E646 80    1C      A      PUT108  SUBA
A E648 2B    A8   E5F2              BMI
A E64A 34    02      A              PSHS
A E64C BD    E762    A              JSR
A E64F 35    02      A              PULS
A E651 48                           LSLA
A E652 8E    EC90    A              LDX
A E655 AD    96      A              JSR
A E657 BD    E76B    A              JSR
A E65A BD    E201    A              JSR
A E65D 0F    20      A              CLR
A E65F 20    91   E5F2              BRA
                            *
A E661 86    20      A      PUTSPC  LDA
             E663    A      PUTCHR  EQU
A E663 34    16      A              PSHS
A E665 A7    9F D045 A              STA
A E669 E6    9F D047 A              LDB
A E66D 2A    09   E678              BPL
A E66F 0D    20      A              TST
A E671 27    05   E678              BEQ
A E673 BD    E6C8    A              JSR
A E676 D6    2C      A              LDB
             E678    A      PUT110  EQU
A E678 BD    E51F    A              JSR
A E67B BD    E79E    A              JSR
A E67E 86    FF      A              LDA
A E680 97    20      A              STA
A E682                              RTSPUL
  A E682 35  96      A              PULS
                            *
                            ***************
                            *
                            * ERASE FIELD
                            *
                            ***************
                            *
             E684    A      PUTOEL  EQU
A E684 0F    20      A              CLR
A E686 BD    E762    A              JSR
A E689 7E    EA9B    A              JMP
                            *
                            ***************
                            *
                            *    BELL
                            *
                            ***************
                            *
             E68C    A      GETBEL  EQU
             E68C    A      PUTOBL  EQU
A E68C                              BUZZON
  A E68C B5  D403    A              BITA
A E68F 39                   PUTNOP  RTS
                            *
                            ***************
                            *
                            *  ERASE ALL
                            *
                            ***************
                            *
             E690    A      PUTOCL  EQU
A E690 96    4D      A              LDA
A E692 BD    E2A5    A              JSR
A E695 0F    20      A              CLR
A E697 39                           RTS
                            *
                            ***************
                            *
                            *  NEW LINE
                            *
                            ***************
                            *
A E698 97    96      A      PUTOCR  STA
A E69A 0F    44      A              CLR
A E69C 96    1F      A              LDA
A E69E 85    20      A              BITA
A E6A0 27    03   E6A5              BEQ
             E6A2    A      PUTOLF  EQU
A E6A2 BD    E7AF    A              JSR
             E6A5    A      PUT550  EQU
A E6A5 0F    20      A              CLR
A E6A7 7E    E201    A      PUT560  JMP
A E6AA 96    4B      A      PUTOHM  LDA
A E6AC 97    43      A              STA
A E6AE 0F    44      A              CLR
A E6B0 20    F3   E6A5              BRA
                            *
                            *
                            ***************
                            *
                            *  START FIELD
                            *
                            ***************
```

```
CURCLR
A
#$14
PUT108  CURSOR ORDER
#PUTORD


[A,X]
PUT010

#$7F
PUT010
PUTCHR
.CURST
PUT010

#$1C    CURSOR CODE?
PUT010  NO, NO OPERATION.
A       SAVE
BTOC
A

#GETJCS CURSOR ENTRY
[A,X]   CURSOR MOVE
CTOB    BUFFER PTR.  SET
BUFPTR
PUTCFG
PUT010

#'
*
A,B,X
[.BPAD] CHAR STORE
[.BAAD] IS HERE TOP OF THE FIELD?
PUT110  NO. (THE FIELD TAG IS THE MSB.)
PUTCFG  CONTINUANCE ?
PUT110  NO.
PUTATR  ATTR SET ONE FIELD
..ACL
*
CDSPB
PUTINC
#$FF
PUTCFG
(A,B,X)
A,B,X,PC

*
*
*
*
*

*
PUTCFG
BTOC
ELKEY

*
*
*
*
*
*
*

BUZZER


*
*
*
*


*
.MDRG+2 KIND OF PLACE
ERASE1
PUTCFG


*
*
*
*
*


PCRORD  SET THE CR ORDER FLAG.
.BPX    SET THE MOST LEFT POSITION.
.CSCTL
#CSLALF IS THE AUTO LF ENABLE?
PUT550  NO.

PUTDW
*
PUTCFG
BUFPTR
.MDRG   TOP Y COORDINATES
.BPY
.BPX
PUT550


***
 *
 *
 *
***
```

```
01403                            *
01404               E6B2       A PUTOSF EQU    *
01405A E6B2 BD     E729        A        JSR    PUTDAT  ATTR GET
01406A E6B5 25     35     E6EC          BCS    PUT720
01407A E6B7 84     3F          A        ANDA   #-(AMFL+AMMD)-$FF01 NOT AMFL
01408A E6B9 97     2C          A        STA    ..ACL
01409A E6BB 8D     0B     E6C8          BSR    PUTATR
01410A E6BD 9E     47          A        LDX    .BAAD
01411A E6BF A6     84          A        LDA    ,X
01412A E6C1 8A     80          A        ORA    #AMFL
01413A E6C3 A7     84          A        STA    ,X
01414A E6C5 0F     20          A        CLR    PUTCFG
01415A E6C7 39                          RTS
01416                            *
01417               E6C8       A PUTATR EQU    *
01418A E6C8 34     06          A        PSHS   A,B
01419A E6CA 9E     47          A        LDX    .BAAD
01420A E6CC A6     84          A        LDA    ,X      LOAD THE ATTR IN CURR. POS.
01421A E6CE 84     3F          A        ANDA   #$3F    PICKUP USER'S ATTR FIELD
01422A E6D0 A7     84          A        STA    ,X
01423A E6D2 91     2C          A        CMPA   ..ACL   SAME AS CURR ATTR
01424A E6D4 27     0C     E6E2          BEQ    PUTAT9  YES, DONT RESTORE.
01425A E6D6 96     21          A        LDA    CSLCDS  MONO?
01426A E6D8 26     08     E6E2          BNE    PUTAT9  YES.  GO RETURN.
01427A E6DA 96     2C          A        LDA    ..ACL
01428               E6DC       A PUTAT5 EQU    *
01429A E6DC A7     80          A        STA    ,X+
01430A E6DE E6     84          A        LDB    ,X
01431A E6E0 2A     FA     E6DC          BPL    PUTAT5
01432A E6E2                      PUTAT9 RTSPUL (A,B)
     A E6E2 35     86          A        PULS   A,B,PC
01433                            *
01434                            ***********************
01435                            *                     *
01436                            *  SET BUFFER ADDRESS *
01437                            *                     *
01438                            ***********************
01439                            *
01440               E6E4       A PUTOSB EQU    *
01441A E6E4 8D     43     E729          BSR    PUTDAT  X CO-ORDINATE
01442A E6E6 25     04     E6EC          BCS    PUT720
01443A E6E8 1F     89          A        TFR    A,B
01444A E6EA 8D     3D     E729          BSR    PUTDAT  Y CO-ORDINATE
01445A E6EC 1025 F9DE E0CE PUT720 LBCS   ERR62
01446A E6F0 1E     89          A        EXG    A,B
01447A E6F2 BD     E3D9        A        JSR    CAXCR
01448A E6F5 1024 F9D5 E0CE        LBCC   ERR62   ORDER ERROR
01449A E6F9 97     44          A        STA    .BPX
01450A E6FB D7     43          A        STB    .BPY
01451A E6FD 8E     D03B        A        LDX    #..DMAP
01452A E700 4F                          CLRA
01453A E701 4C                   PUT750 INCA
01454A E702 E1     81          A        CMPB   ,X++
01455A E704 25     FB     E701          BLO    PUT750
01456A E706 E1     1F          A        CMPB   -1,X
01457A E708 24     F7     E701          BHS    PUT750
01458A E70A 97     4D          A        STA    .MDRG+2 KIND
01459A E70C EC     1E          A        LDD    -2,X
01460A E70E DD     4B          A        STD    .MDRG   RANGE
01461A E710 BD     E92F        A        JSR    GETRNG
01462A E713 20     90     E6A5          BRA    PUT550
01463                            *
01464                            **********************
01465                            *                    *
01466                            * REPEAT CHARACTER   *
01467                            *                    *
01468                            **********************
01469                            *
01470               E715       A PUTORC EQU    *
01471A E715 8D     12     E729          BSR    PUTDAT  NO. OF REP
01472A E717 25     D3     E6EC          BCS    PUT720
01473A E719 1F     89          A        TFR    A,B
01474A E71B 8D     0C     E729          BSR    PUTDAT  CHAR
01475A E71D 25     CD     E6EC          BCS    PUT720
01476A E71F 5D                          TSTB
01477A E720 27     06     E728          BEQ    PUT809
01478A E722 BD     E663        A PUT805 JSR    PUTCHR
01479A E725 5A                          DECB
01480A E726 26     FA     E722          BNE    PUT805
01481A E728 39                   PUT809 RTS
01482                            *
01483               E729       A PUTDAT EQU    *
01484A E729 34     10          A        PSHS   X
01485               E72B       A PUTDA1 EQU    *
01486A E72B B6     D383        A        LDA    .COMAR+3
01487A E72E 26     08     E738          BNE    PUTDA2
01488A E730 96     02          A        LDA    ..CONT
01489A E732 2B     11     E745          BMI    PUTDA3
01490A E734 1A     01                   SEC
01491A E736                             RTSPUL X
     A E736 35     90          A        PULS   X,PC
01492A E738 9E     98          A PUTDA2 LDX    PUTIX
01493A E73A A6     80          A        LDA    ,X+
01494A E73C 9F     98          A        STX    PUTIX
01495A E73E 7A     D383        A        DEC    .COMAR+3
01496A E741 1C     FE                   CLC
01497A E743                             RTSPUL X
     A E743 35     90          A        PULS   X,PC
01498A E745 BD     E11A        A PUTDA3 JSR    CONTI
01499A E748 8E     D3B4        A        LDX    #.COMAR+4
01500A E74B 9F     98          A        STX    PUTIX
01501A E74D 20     DC     E72B          BRA    PUTDA1
01502
01503                            ***************
01504                            *             *
01505                            *  BACKSPACE  *
01506                            *             *
01507                            ***************
```

```
                          *
            E74F     A PUTOBS EQU    *
A E74F 9E   47       A        LDX    .BAAD
A E751 A6   84       A        LDA    ,X
A E753 2B   0A  E75F          BMI    PUTOB9  FIELD TOP
A E755 8D   0B  E762          BSR    BTOC
A E757 BD   EA0F     A        JSR    CSL
A E75A 8D   0F  E76B          BSR    CTOB
A E75C BD   E201     A        JSR    BUFPTR
A E75F 0F   20       A PUTOB9 CLR    PUTCFG
A E761 39                     RTS
                          *
A E762 8E   D043     A BTOC   LDX    #.BPY
A E765 108E D04E     A        LDY    #.CY
A E769 20   07  E772          BRA    CTOBM
A E76B 8E   D04E     A CTOB   LDX    #.CY
A E76E 108E D043     A        LDY    #.BPY
A E772 C6   08       A CTOBM  LDB    #8
A E774 7E   E44E     A        JMP    MOVE
                          *
                          *********
                          *       *
                          *  TAB  *
                          *       *
                          *********
                          *
            E777     A PUTOTB EQU    *
A E777 BD   E661     A        JSR    PUTSPC
A E77A 96   44       A        LDA    .BPX
A E77C 8D   03  E781          BSR    TABCHK
A E77E 27   F7  E777          BEQ    PUTOTB
A E780 39                     RTS
                          *
            E781     A TABCHK EQU    *
A E781 D6   1F       A        LDB    .CSCTL
A E783 C5   04       A        BITB   #CSLTAB
A E785 27   15  E79C          BEQ    TABCK9
A E787 4D                     TSTA
A E788 27   12  E79C          BEQ    TABCK9
A E78A 1F   89       A        TFR    A,B
A E78C 54                     LSRB
A E78D 54                     LSRB
A E78E 54                     LSRB
A E78F 84   07       A        ANDA   #$07
A E791 8E   E396     A        LDX    #BAS
A E794 A6   86       A        LDA    A,X
A E796 8E   D022     A        LDX    #.TABST
A E799 A5   85       A        BITA   B,X
A E79B 39                     RTS
A E79C 4C              TABCK9 INCA
A E79D 39                     RTS
                          *
                          ***************
                          *             *
                          *  INCREMANT  *
                          *             *
                          ***************
                          *
            E79E     A PUTINC EQU    *
A E79E 0C   44       A        INC    .BPX
A E7A0 96   44       A        LDA    .BPX
A E7A2 91   34       A        CMPA   ..LNC
A E7A4 24   03  E7A9          BHS    PUTIN5  RIGHT OVER
            E7A6     A PUTIN3 EQU    *
A E7A6 7E   E201     A        JMP    BUFPTR
                          *
A E7A9 0F   44       A PUTIN5 CLR    .BPX    DOWN
A E7AB 96   1F       A        LDA    .CSCTL
A E7AD 2B   F7  E7A6          BMI    PUTIN3
                          *
                          ***************
                          *             *
                          *   DOWN      *
                          *             *
                          ***************
                          *
            E7AF     A PUTDW  EQU    *
A E7AF 0C   43       A        INC    .BPY
A E7B1 96   43       A        LDA    .BPY
A E7B3 91   4C       A        CMPA   .MDRG+1
A E7B5 25   EF  E7A6          BLO    PUTIN3  INNER
                          *
A E7B7 96   4D       A        LDA    .MDRG+2 KIND OF WINDOW
A E7B9 81   01       A        CMPA   #1
A E7BB 27   1D  E7DA          BEQ    SCROLL
                          * PAGE
A E7BD 96   1F       A        LDA    .CSCTL
A E7BF 85   08       A        BITA   #CSLPAG PAGE LIMIT WAIT ?
A E7C1 27   14  E7D7          BEQ    PUTD70  NO.
A E7C3                        BUZZON
  A E7C3 B5 D403     A        BITA   BUZZER
A E7C6 8E   E690     A        LDX    #PUTOCL
A E7C9 9F   00       A        STX    ..IRQE  ENTRY SET
A E7CB 1C   EF                CLI            IRQ ENABLE
A E7CD BD   FD8E     A        JSR    KEYGET
A E7D0 1A   10                SEI            IRQ MASK
A E7D2 8E   0000     A        LDX    #0
A E7D5 9F   00       A        STX    ..IRQE  ENTRY CLEAR
A E7D7 7E   E690     A PUTD70 JMP    PUTOCL
                          ************
                          *          *
                          *  SCROLL  *
                          *          *
                          ************
                          *
            E7DA     A SCROLL EQU    *
A E7DA 96   4C       A        LDA    .MDRG+1
A E7DC D6   4C       A        LDB    .MDRG+1
A E7DE 5C                     INCB
A E7DF DD   73       A        STD    .CVTOP
A E7E1 91   41       A        CMPA   ..DMAP+6
A E7E3 26   32  E817          BNE    SCR004
A E7E5 D6   4B       A        LDB    .MDRG
                          *
A E7E7 34   01       A        PSHS   CC
A E7E9 26   2E  E819          BNE    SCR005
A E7EB 96   36       A        LDA    ..FS
A E7ED 27   11  E800          BEQ    SCR002
A E7EF DC   73       A        LDD    .CVTOP
A E7F1 C3   0202     A        ADDD   #$0202
A E7F4 DD   73       A        STD    .CVTOP
A E7F6 BD   E2D7     A        JSR    ERSVRM
A E7F9 DC   73       A        LDD    .CVTOP
A E7FB 83   0202     A        SUBD   #$0202
A E7FE DD   73       A        STD    .CVTOP
            E800     A SCR002 EQU    *
A E800 BD   E2D7     A        JSR    ERSVRM
A E803 DC   1D       A        LDD    .VOFST
A E805 D3   39       A        ADDD   ..VLU
A E807                        VACON
  A E807 B5 D409     A        BITA   VACC
A E80A DD   1D       A        STD    .VOFST
A E80C FD   D40E     A        STD    SOFFST
A E80F                        VACOFF
  A E80F B7 D409     A        STA    VACC
A E812 BD   EDBE     A        JSR    INDICT
A E815 20   51  E868          BRA    SCR015
                          *
            E817     A SCR004 EQU    *
A E817 34   01       A        PSHS   CC
A E819 4A              SCR005 DECA
A E81A 91   4B       A        CMPA   .MDRG
A E81C 27   6A  E888          BEQ    SCR090
A E81E 96   4C       A        LDA    .MDRG+1
A E820 BD   E226     A        JSR    BVADL
A E823 1F   01       A        TFR    D,X
A E825 96   4B       A        LDA    .MDRG
A E827 BD   E226     A        JSR    BVADL
A E82A 1F   02       A        TFR    D,Y
A E82C D3   39       A        ADDD   ..VLU
A E82E 1F   03       A        TFR    D,U
A E830                        VACON
  A E830 B5 D409     A        BITA   VACC
A E833 96   21       A        LDA    CSLCDS
A E835 26   20  E857          BNE    MNSCRL
A E837 34   10       A        PSHS   X
A E839 1F   31       A        TFR    U,X
A E83B EC   89 8000  A MRSCRL LDD    ..VG,X
A E83F ED   A9 8000  A        STD    ..VG,Y
A E843 EC   89 4000  A        LDD    ..VR,X
A E847 ED   A9 4000  A        STD    ..VR,Y
A E84B EC   81       A        LDD    ,X++
A E84D ED   A1       A        STD    ,Y++
A E84F AC   E4       A        CMPX   ,S
A E851 25   E8  E83B          BLO    MRSCRL
A E853 32   62       A        LEAS   2,S
A E855 20   0E  E865          BRA    SCR013
A E857 30   89 8000  A MNSCRL LEAX   ..VG,X
A E85B 31   A9 8000  A        LEAY   ..VG,Y
A E85F 33   C9 8000  A        LEAU   ..VG,U
A E863 8D   38  E89D          BSR    COPLOP
A E865                 SCR013 VACOFF
  A E865 B7 D409     A        STA    VACC
                          *
A E868 96   4B       A SCR015 LDA    .MDRG
A E86A BD   E20D     A        JSR    BADRL
A E86D 34   30       A        PSHS   X,Y
A E86F 96   4C       A        LDA    .MDRG+1
A E871 BD   E20D     A        JSR    BADRL
A E874 35   40       A        PULS   U
A E876 34   20       A        PSHS   Y
A E878 31   C4       A        LEAY   ,U
A E87A D6   34       A        LDB    ..LNC
A E87C 33   C5       A        LEAU   B,U
A E87E 8D   1D  E89D          BSR    COPLOP
A E880 35   30       A        PULS   X,Y
                          *
A E882 D6   34       A        LDB    ..LNC
A E884 33   A5       A        LEAU   B,Y
A E886 8D   15  E89D          BSR    COPLOP
                          *
            E888     A SCR090 EQU    *
A E888 DC   73       A        LDD    .CVTOP
A E88A 4A                     DECA
A E88B 5A                     DECB
A E88C 97   43       A        STA    .BPY
A E88E DD   73       A        STD    .CVTOP
A E890 35   01       A        PULS   CC
A E892 27   03  E897          BEQ    SCR095
A E894 BD   E2D7     A        JSR    ERSVRM
A E897 BD   E335     A SCR095 JSR    ERSBUF
A E89A 7E   E2C9     A        JMP    ASMFLD
                          *
            E89D     A COPLOP EQU    *
                          *
                          * COPY DATA
                          *
                          * U SOURCE TOP ADDRESS
                          * Y DESTINATION TOP AD
                          * X SOURCE END ADDRESS
                          *
                          * U .LT. X
                          * MOD (X-U,8) .EQ. 0
                          *
A E89D 34   10       A        PSHS   X
```

```
IS THE DISPLAY FULL SCROLL?
NO.
IS THE TOP OF SCROLL WINDOW SAME AS THE TOP OF
 DISPLAY?

NO.
DISPLAY PF?
NO.
YES, ERASE THE LAST LINE OF DISPLAY.

ERASE LAST LINE BEFOR SCROLLING.

ONLY ONE LINE

SOURCE END ADDRESS

DESTINATION TOP

SOURCE TOP

MONO?
YES.
SOURCE END
LOAD TOP OF SOURCE TO X.

DESTINATION TOP ADDRESS

SAVE
SOURCE END ADDRESS

LOAD DESTINATION TOP ADDRESS.
SAVE ATTR'S SOURCE END ADDRESS.
SET Y DESTINATION TOP ADDRESS.

SET U SOURCE TOP ADDRESS
COPY THE CHAR.
LOAD SORCE ADDRESS TO X.
 LOAD DESTINATION TOP ADDRESS TO Y.

MAKE SOURCE TOP ADDRESS.
COPY THE ATTR.

ASSUME FIELD.

DRESS
```

```
A E89F 37    16        A COPLDO PULU   D,X
A E8A1 ED    A4        A        STD    ,Y
A E8A3 AF    22        A        STX    2,Y
A E8A5 37    16        A        PULU   D,X
A E8A7 ED    24        A        STD    4,Y
A E8A9 AF    26        A        STX    6,Y
A E8AB 31    28        A        LEAY   8,Y
A E8AD 11A3  E4        A        CMPU   ,S
A E8B0 25    ED   E89F          BLO    COPLDO
A E8B2                          RTSPUL X
  A E8B2 35  90        A        PULS   X,PC

               001B    A ESC    EQU    $1B
                                *
               E8B4    A PUTORD EQU    *
A E8B4         E68F    A        FDB    PUTNOP  $00
A E8B6         E68F    A        FDB    PUTNOP  $01
A E8B8         E68F    A        FDB    PUTNOP  $02
A E8BA         E68F    A        FDB    PUTNOP  $03
A E8BC         E68F    A        FDB    PUTNOP  $04
A E8BE         E6B4    A        FDB    PUTOEL  $05
A E8C0         E68F    A        FDB    PUTNOP  $06
A E8C2         E68C    A        FDB    PUTOBL  $07
A E8C4         E74F    A        FDB    PUTOBS  $08
A E8C6         E777    A        FDB    PUTOTB  $09
A E8C8         E6A2    A        FDB    PUTOLF  $0A
A E8CA         E6AA    A        FDB    PUTOHM  $0B
A E8CC         E690    A        FDB    PUTOCL  $0C
A E8CE         E69B    A        FDB    PUTOCR  $0D
A E8D0         E68F    A        FDB    PUTNOP  $0E
A E8D2         E68F    A        FDB    PUTNOP  $0F
A E8D4         E68F    A        FDB    PUTNOP  $10
A E8D6         E6B2    A        FDB    PUTOSF  $11
A E8D8         E6E4    A        FDB    PUTOSB  $12
A E8DA         E715    A        FDB    PUTORC  $13
                                *
                                TTL    GETCM - GET CODE -
                                ****************
                                *              *
                                * GET COMMAND  *
                                *              *
                                ****************
                                *
               D09B    A GTTOPC EQU    PUTEND
               D09D    A GTBTMC EQU    GTTOPC+2
               D09F    A GTTOPA EQU    GTBTMC+2
               D0A1    A GTBTMA EQU    GTTOPA+2
               D0A3    A GTFLD  EQU    GTBTMA+2
               D0A4    A GTSCRL EQU    GTFLD+1
                                *
               E8DC    A GET000 EQU    *
A E8DC 0F    20        A        CLR    PUTCFG
A E8DE BD    FDDE      A        JSR    KEYR2   DUMMY READ
A E8E1 BD    ED94      A        JSR    CURSET
A E8E4 8E    E927      A        LDX    #GETABT
A E8E7 9F    00        A        STX    ..IRQE
                                *
                                * REMOVE MODIFIED DATA TAG
                                *
A E8E9 9E    9F        A        LDX    GTTOPA
A E8EB A6    84        A GET050 LDA    ,X      ATTR
A E8ED 84    BF        A        ANDA   #-AMMD-$FF01 REMOVE
A E8EF A7    80        A        STA    ,X+
A E8F1 9C    A1        A        CMPX   GTBTMA
A E8F3 25    F6   E8EB          BLO    GET050
                                *
               E8F5    A GET100 EQU    *
A E8F5 BD    E3C7      A        JSR    CBADR
A E8F8 86    01        A        LDA    #1
A E8FA 97    1B        A        STA    .CURTM
A E8FC BD    EDB7      A        JSR    CURGO
A E8FF 1C    EF                 CLI            IRQ ENABLE
A E901 BD    FDBE      A        JSR    KEYGET
A E904 1A    10                 SEI            IRQ MASK
A E906 34    02        A        PSHS   A
A E908 BD    ED9D      A        JSR    CURSCL
A E90B 35    02        A        PULS   A
A E90D 81    7F        A        CMPA   #$7F    DELETE
A E90F 26    05   E916          BNE    GET110
A E911 BD    EB97      A        JSR    GETDEL
A E914 20    DF   E8F5          BRA    GET100
A E916 81    20        A GET110 CMPA   #'
A E918 25    05   E91F          BLO    GET120
A E91A BD    E944      A        JSR    GETDT   DATA
A E91D 20    D6   E8F5          BRA    GET100
A E91F 48                GET120 LSLA
A E920 8E    EC58      A        LDX    #GETJMP CONTROL KEY
A E923 AD    96        A        JSR    [A,X]
A E925 20    CE   E8F5          BRA    GET100
A E927 8E    0000      A GETABT LDX    #0
A E92A 9F    00        A        STX    ..IRQE
A E92C 7E    ED9D      A        JMP    CURSCL
                                *
                                * WINDOW RANGE
                                *
                                *  X CHAR TOP
                                *  Y ATTR TOP
                                *
               E92F    A GETRNG EQU    *
A E92F 96    4C        A        LDA    .MDRG+1
A E931 BD    E20D      A        JSR    BADRL
A E934 9F    9D        A        STX    GTBTMC  CHAR BTM
A E936 109F  A1        A        STY    GTBTMA  ATTR BTM
A E939 96    4B        A        LDA    .MDRG
A E93B BD    E20D      A        JSR    BADRL
A E93E 9F    9B        A        STX    GTTOPC
A E940 109F  9F        A        STY    GTTOPA
```

```
01826A E943 39           GETNOP RTS
01827                    *
01828                    **********
01829                    *        *
01830                    *  DATA  *
01831                    *        *
01832                    **********
01833                    *
01834         E944     A GETDT  EQU   *
01835A E944 9E   50    A        LDX   .CBAD
01836A E946 109E 52    A        LDY   .CAAD
01837A E949 E6   A4    A        LDB   ,Y
01838A E94B C5   10    A        BITB  #AMPT
01839A E94D 27   04  E953       BEQ   GETDTO
01840A E94F                     BUZZON
     A E94F B5   D403  A        BITA  BUZZER
01841A E952 39                  RTS
01842
01843A E953 0D   31    A GETDTO TST   ..INS    INSERT ?
01844A E955 26   35  E98C       BNE   GETINS   YES
01845A E957 A7   84    A GETDT1 STA   ,X
01846A E959 CA   40    A        ORB   #AMMD
01847A E95B E7   A4    A        STB   ,Y
01848A E95D BD   E51C  A        JSR   CDSPC
01849A E960 BD   E9F3  A        JSR   CSR
01850A E963 24   14  E979       BCC   GETDT2
01851A E965 96   4D    A        LDA   .MDRG+2
01852A E967 4A                  DECA
01853A E968 26   0F  E979       BNE   GETDT2   PAGE
01854A E96A BD   E7DA  A        JSR   SCROLL
01855A E96D 03   A4    A        COM   GTSCRL   SET SCROLL FLAG FOR INSERT MODE.
01856A E96F 0F   44    A        CLR   .BPX
01857A E971 DC   43    A        LDD   .BPY
01858A E973 DD   4E    A        STD   .CY
01859A E975 7E   E201  A        JMP   BUFPTR
01860A E978 39           GETDT4 RTS
01861A E979 BD   E3C7  A GETDT2 JSR   CBADR
01862A E97C E6   A4    A        LDB   ,Y
01863A E97E 2A   03  E983       BPL   GETDT3
01864A E980                     BUZZON
     A E980 B5   D403  A        BITA  BUZZER
01865A E983 9E   4E    A GETDT3 LDX   .CY
01866A E985 9C   43    A        CMPX  .BPY
01867A E987 23   EF  E978       BLS   GETDT4
01868A E989 7E   E76B  A        JMP   CTOB
01869                    ;
01870         E98C     A GETINS EQU   *
01871A E98C 0F   A4    A        CLR   GTSCRL
01872A E98E 31   21    A        LEAY  1,Y      PRE INC
01873A E990 6D   80    A GETI10 TST   ,X+      NULL SEARCH
01874A E992 27   0B  E99C       BEQ   GETI20   FOUND
01875A E994 6D   A0    A        TST   ,Y+      AMFL IS MSB
01876A E996 2A   F8  E990       BPL   GETI10   NOT FIELD
01877A E998                     BUZZON
     A E998 B5   D403  A        BITA  BUZZER
01878A E99B 39                  RTS
01879         E99C     A GETI20 EQU   *
01880A E99C 9E   4E    A        LDX   .CY
01881A E99E 34   10    A        PSHS  X
01882A E9A0 9E   50    A        LDX   .CBAD
01883A E9A2 109E 52    A        LDY   .CAAD
01884         E9A5     A GETI30 EQU   *
01885A E9A5 1F   03    A        TFR   D,U      CHAR,ATTR SAVE
01886A E9A7 A6   84    A        LDA   ,X
01887A E9A9 E6   A4    A        LDB   ,Y
01888A E9AB C4   7F    A        ANDB  #-AMFL-$FF01 REMOVE FIELD
01889A E9AD 34   06    A        PSHS  A,B      STORE POS DATA SAVE
01890A E9AF 1F   30    A        TFR   U,D
01891A E9B1 8D   A4  E957       BSR   GETDT1
01892A E9B3 BD   E3C7  A        JSR   CBADR
01893A E9B6 35   06    A        PULS  A,B
01894A E9B8 4D                  TSTA
01895A E9B9 26   EA  E9A5       BNE   GETI30   NOT NULL
01896A E9BB 35   06    A        PULS  D
01897A E9BD 9B   A4    A        ADDA  GTSCRL
01898A E9BF DD   4E    A        STD   .CY
01899A E9C1 20   30  E9F3       BRA   CSR
01900                    *
01901                    ****************
01902                    *              *
01903                    *  RIGHT.WRAP  *
01904                    *              *
01905                    ****************
01906                    *
01907         E9C3     A CSRL   EQU   *
01908A E9C3 0C   4F    A        INC   .CX
01909A E9C5 96   4F    A        LDA   .CX
01910A E9C7 91   34    A        CMPA  ..LNC
01911A E9C9 2D   02  E9CD       BLT   CSRL90
01912A E9CB 0F   4F    A        CLR   .CX      OVER
01913A E9CD 39           CSRL90 RTS
01914                    *
01915                    ***************
01916                    *             *
01917                    *  LEFT.WRAP  *
01918                    *             *
01919                    ***************
01920                    *
01921         E9CE     A CSLL   EQU   *
01922A E9CE 0A   4F    A        DEC   .CX
01923A E9D0 2A   05  E9D7       BPL   CSLL90
01924A E9D2 96   34    A        LDA   ..LNC
01925A E9D4 4A                  DECA
01926A E9D5 97   4F    A        STA   .CX
01927A E9D7 39           CSLL90 RTS
01928                    *
01929                    **************
```

```
01930                    *
01931                    *  UP.
01932                    *
01933                    ******
01934                    *
01935         E9D8     A CSUL
01936A E9D8 0A   4E    A
01937A E9DA 96   4E    A
01938A E9DC 91   4B    A
01939A E9DE 2C   05  E9E5
01940A E9E0 96   4C    A
01941A E9E2 4A
01942A E9E3 97   4E    A
01943A E9E5 39           CSUL90
01944                    *
01945                    ******
01946                    *
01947                    *  DOW
01948                    *
01949                    ******
01950                    *
01951         E9E6     A CSDL
01952A E9E6 0C   4E    A
01953A E9E8 96   4E    A
01954A E9EA 91   4C    A
01955A E9EC 2D   04  E9F2
01956A E9EE 96   4B    A
01957A E9F0 97   4E    A
01958A E9F2 39           CSDL90
01959                    *
01960                    ******
01961                    *
01962                    *  RIG
01963                    *
01964                    ******
01965                    *
01966         E9F3     A CSR
01967A E9F3 0C   4F    A
01968A E9F5 96   4F    A
01969A E9F7 91   34    A
01970A E9F9 2D   11  EA0C
01971A E9FB 0F   4F    A
01972A E9FD 0C   4E    A
01973A E9FF 96   4E    A
01974A EA01 91   4C    A
01975A EA03 2D   07  EA0C
01976A EA05 96   4B    A
01977A EA07 97   4E    A
01978A EA09 1A   01
01979A EA0B 39
01980                    *
01981         EA0C     A CSL90
01982A EA0C 1C   FE      CSR90
01983A EA0E 39
01984                    *
01985                    ******
01986                    *
01987                    *  LEF
01988                    *
01989                    ******
01990                    *
01991         EA0F     A CSL
01992A EA0F 0A   4F    A
01993A EA11 2A   F9  EA0C
01994A EA13 96   34    A
01995A EA15 4A
01996A EA16 97   4F    A
01997A EA18 0A   4E    A
01998A EA1A 96   4E    A
01999A EA1C 91   4B    A
02000A EA1E 2C   EC  EA0C
02001A EA20 96   4C    A
02002A EA22 4A
02003A EA23 97   4E    A
02004A EA25 1A   01
02005A EA27 39
02006                    *
02007                    *
02008                    ******
02009                    *
02010                    *  ENC
02011                    *
02012                    ******
02013                    *
02014         EA28     A GETENC
02015A EA28 1F   10    A
02016A EA2A 93   9B    A
02017A EA2C 20   04  EA32
02018                    *
02019         EA2E     A GETENA
02020A EA2E 1F   20    A
02021A EA30 93   9F    A
02022                    *
02023         EA32     A GETEN
02024A EA32 9E   4B    A
02025A EA34 9F   4E    A
02026                    *
02027A EA36 93   33    A GETEN5
02028A EA38 2B   04  EA3E
02029A EA3A 0C   4E    A
02030A EA3C 20   F8  EA36
02031                    *
02032A EA3E D3   33    A GETEN9
02033A EA40 D7   4F    A
02034A EA42 39
02035                    *
02036                    ******
02037                    *
```

```
       LEFT     *
WRAP            *
                *
*******
EQU   *
DEC   .CY
LDA   .CY
CMPA  .MDRG
BGE   CSUL90
LDA   .MDRG+1
DECA
STA   .CY
RTS

*********
                *
N.WRAP          *
                *
*********

EQU   *
INC   .CY
LDA   .CY
CMPA  .MDRG+1
BLT   CSDL90
LDA   .MDRG
STA   .CY
RTS

*****
    *
HT  *
    *
*****

EQU   *
INC   .CX
LDA   .CX
CMPA  ..LNC
BLT   CSR90
CLR   .CX
INC   .CY
LDA   .CY
CMPA  .MDRG+1
BLT   CSR90
LDA   .MDRG
STA   .CY
SEC
RTS

EQU   *
CLC
RTS

****
   *
T  *
   *
****

EQU   *
DEC   .CX
BPL   CSL90
LDA   ..LNC
DECA
STA   .CX
DEC   .CY
LDA   .CY
CMPA  .MDRG
BGE   CSL90
LDA   .MDRG+1
DECA
STA   .CY
SEC
RTS

******************
                 *
ODE CO-ORDINATE  *
                 *
******************

EQU   *
TFR   X,D      CHAR
SUBD  GTTOPC   OFFSET
BRA   GETEN

EQU   *
TFR   Y,D
SUBD  GTTOPA   OFFSET

EQU   *
LDX   .MDRG
STX   .CY      TOP Y,DESTROY .CX

SUBD  ..LNCD
BMI   GETEN9
INC   .CY
BRA   GETEN5

ADDD  ..LNCD
STB   .CX
RTS

****
   *
```

```
02038                                *     HOME   *
02039                                *            *
02040                                ***********
02041                                *
02042                   EA43     A CSHOME EQU   *
02043A EA43 109E 9F         A          LDY   GTTOPA   ATTR
02044A EA46 86    B0        A CSHOM0 LDA   #AMFL
02045A EA48 C6    10        A          LDB   #AMPT
02046A EA4A A5    A4        A CSHOM5 BITA  ,Y
02047A EA4C 27    04   EA52            BEQ   CSHOM7
02048A EA4E E5    A4        A          BITB  ,Y
02049A EA50 27    0A   EA5C            BEQ   CSHOM9
02050A EA52 31    21        A CSHOM7 LEAY  1,Y
02051A EA54 109C A1         A          CMPY  GTBTMA
02052A EA57 25    F1   EA4A            BLO   CSHOM5
02053A EA59 109E 9F         A CSHOM8 LDY   GTTOPA
02054A EA5C BD    D0   EA2E CSHOM9 BSR   GETENA
02055A EA5E 39                         RTS
02056                                *
02057                                *******************
02058                                *                 *
02059                                *   FORWARD FIELD *
02060                                *                 *
02061                                *******************
02062                                *
02063                   EA5F     A GETFFD EQU   *
02064A EA5F BD    E9F3      A          JSR   CSR      RIGHT
02065A EA62 BD    E3C7      A          JSR   CBADR
02066A EA65 20    DF   EA46            BRA   CSHOM0
02067                                *
02068                                *
02069                                *   BACKWARD FIELD
02070                                *
02071                   EA67     A GETBFD EQU   *
02072A EA67 BD    EA0F      A          JSR   CSL      LEFT
02073A EA6A BD    E3C7      A          JSR   CBADR
02074                                *
02075                   EA6D     A GETBHM EQU   *
02076                                *
02077                                *   SEARCH UNPROTECTED FIELD
02078                                *   FOR BACK
02079                                *
02080A EA6D CC    B010      A          LDD   #AMFL*256+AMPT
02081A EA70 A5    A4        A GETBH5 BITA  ,Y
02082A EA72 27    04   EA78            BEQ   GETBH7   NO FIELD
02083A EA74 E5    A4        A          BITB  ,Y
02084A EA76 27    E4   EA5C            BEQ   CSHOM9   DEFINE
02085A EA78 31    3F        A GETBH7 LEAY  -1,Y
02086A EA7A 109C 9F         A          CMPY  GTTOPA
02087A EA7D 24    F1   EA70            BHS   GETBH5
02088A EA7F 20    D8   EA59            BRA   CSHOM8   PHISICAL HOME
02089                                *
02090                                **********
02091                                *        *
02092                                *  CLEAR *
02093                                *        *
02094                                **********
02095                                *
02096                   EAB1     A GETCLR EQU   *
02097A EA81 96    4D        A          LDA   .MDRG+2
02098A EA83 BD    E2A5      A          JSR   ERASE1
02099A EA86 DC    43        A          LDD   .BPY
02100A EA88 DD    4E        A          STD   .CY
02101A EA8A 86    02        A          LDA   #2
02102A EA8C 7E    EBEB      A          JMP   GETCR1
02103                                *
02104                                ***********
02105                                *         *
02106                                *  INSERT *
02107                                *         *
02108                                ***********
02109                                *
02110                   EA8F     A INSERT EQU   *
02111A EA8F 03    31        A          COM   ..INS
02112A EA91 26    04   EA97            BNE   INSER5
02113A EA93                            INSOFF
     A EA93 B7    D40D      A          STA   INSLED
02114A EA96 39                         RTS
02115A EA97                     INSER5 INSON
     A EA97 B5    D40D      A          BITA  INSLED
02116A EA9A 39                         RTS
02117                                *
02118                                *********************
02119                                *                   *
02120                                *  ERASE FIELD KEY  *
02121                                *                   *
02122                                *********************
02123                                *
02124                   EA9B     A ELKEY  EQU   *
02125A EA9B DC    4E        A          LDD   .CY
02126A EA9D 34    06        A          PSHS  D
02127A EA9F 9E    50        A          LDX   .CBAD
02128A EAA1 109E 52         A          LDY   .CAAD
02129A EAA4 E6    A4        A          LDB   ,Y
02130A EAA6 CA    40        A          ORB   #AMMD    MODIFY TAG
02131A EAA8 E7    A4        A          STB   ,Y
02132                   EAAA     A ELKEY2 EQU   *
02133A EAAA A6    84        A          LDA   ,X
02134A EAAC 27    08   EAB6            BEQ   ELKEY4   NULL SKIP
02135A EAAE 4F                         CLRA
02136A EAAF A7    84        A          STA   ,X
02137A EAB1 E6    A4        A          LDB   ,Y
02138A EAB3 BD    E51C      A          JSR   CDSPC
02139                   EAB6     A ELKEY4 EQU   *
02140A EAB6 BD    E9F3      A          JSR   CSR
02141A EAB9 BD    E3C7      A          JSR   CBADR
02142A EABC E6    A4        A          LDB   ,Y       IS HERE THE TOP OF FIELD?
```

```
02143A EABE 2A    EA   EAAA        BPL    ELKEY2  NO. (FIELD FLAG IS MSB)
02144A EAC0 9C    45      A        CMPX   .BPAD   IS HERE OVER THE BUFFER POINTER?
02145A EAC2 23    07   EACB        BLS    ELKEY6
02146A EAC4 EC    E4      A        LDD    ,S
02147A EAC6 DD    43      A        STD    .BPY
02148A EAC8 BD    E201    A        JSR    BUFFPTR
02149             EACB    A ELKEY6 EQU    *
02150A EACB 35    06      A        PULS   D
02151A EACD DD    4E      A        STD    .CY
02152A EACF 39                     RTS
02153                       *
02154                       *********
02155                       *       *
02156                       *  TAB  *
02157                       *       *
02158                       *********

02160             EAD0    A GETDUP EQU    *       DUPLICATE
02161A EAD0 96    34      A        LDA    ..LNC
02162A EAD2 D6    4E      A        LDB    .CY
02163A EAD4 D1    4B      A        CMPB   .MDRG
02164A EAD6 27    2C   EB04        BEQ    GETDU9
02165A EAD8 40                     NEGA
02166A EAD9                        SKIP1
     A EAD9       21      A        FCB    $21
02167                       *
02168             EADA    A TAB    EQU    *
02169A EADA 4F                     CLRA
02170A EADB D6    31      A TABDUP LDB    ..INS
02171A EADD 34    06      A        PSHS   D
02172A EADF 0F    31      A        CLR    ..INS   UNEFFECT
02173A EAE1 86    20      A TAB10  LDA    #'
02174A EAE3 E6    E4      A        LDB    ,S      IS IT DUPLICATE?
02175A EAE5 27    04   EAEB        BEQ    TAB15   NO, TAB.
02176A EAE7 9E    50      A        LDX    .CBAD
02177A EAE9 A6    85      A        LDA    B,X
02178A EAEB BD    E944    A TAB15  JSR    GETDT
02179A EAEE BD    E3C7    A        JSR    CBADR
02180A EAF1 96    4F      A        LDA    .CX
02181A EAF3 BD    E781    A        JSR    TABCHK  TAB STOP?
02182A EAF6 26    08   EB00        BNE    TAB90   YES. (TAB DISABLE)
02183A EAF8 9E    52      A        LDX    .CAAD   NO.
02184A EAFA E6    84      A        LDB    ,X      FIELD TOP
02185A EAFC C5    90      A        BITB   #AMFL+AMPT OR PROTECT?
02186A EAFE 27    E1   EAE1        BEQ    TAB10   NO.
02187A EB00 35    06      A TAB90  PULS   D
02188A EB02 D7    31      A        STB    ..INS
02189A EB04 39            GETDU9   RTS
02190                       *
02191                       *****************
02192                       *               *
02193                       *  RIGHT WORD   *
02194                       *               *
02195                       *****************
02196                       *
02197             EB05    A CSRW   EQU    *
02198A EB05 0F    A3      A        CLR    GTFLD   FIELD FOUND CLEAR
02199A EB07 9E    50      A        LDX    .CBAD
02200A EB09 109E  52      A        LDY    .CAAD
02201A EB0C 8D    16   EB24 CSRW10 BSR    CSRWS
02202A EB0E 25    FC   EB0C        BCS    CSRW10  ON
02203                       *
02204A EB10 8D    12   EB24 CSRW20 BSR    CSRWS
02205A EB12 24    FC   EB10        BCC    CSRW20  OFF
02206A EB14 30    1F      A        LEAX   -1,X    STOP POS
02207                       *
02208                       * ON
02209                       *
02210A EB16 9C    9D      A CSRW90 CMPX   GTBTMC
02211A EB18 25    07   EB21        BLO    CSRW95
02212A EB1A 96    4B      A        LDA    .MDRG   OVER
02213A EB1C 97    4E      A        STA    .CY
02214A EB1E 0F    4F      A        CLR    .CX
02215A EB20 39                     RTS
02216A EB21 7E    EA28    A CSRW95 JMP    GETENC
02217                       *
02218             EB24    A CSRWS  EQU    *
02219A EB24 0D    A3      A        TST    GTFLD
02220A EB26 26    0F   EB37        BNE    CSRWS9
02221A EB28 8D    11   EB3B        BSR    CSWCK
02222A EB2A 30    01      A        LEAX   1,X
02223A EB2C 31    21      A        LEAY   1,Y
02224A EB2E C6    80      A        LDB    #AMFL
02225A EB30 E5    A4      A        BITB   ,Y
02226A EB32 27    02   EB36        BEQ    CSRWS5
02227A EB34 0C    A3      A        INC    GTFLD
02228A EB36 39            CSRWS5   RTS
02229A EB37 32    62      A CSRWS9 LEAS   2,S
02230A EB39 20    DB   EB16        BRA    CSRW90
02231                       *
02232             EB3B    A CSWCK  EQU    *
02233A EB3B A6    84      A        LDA    ,X
02234A EB3D CE    EB81    A        LDU    #CSWCKT
02235A EB40 E6    C0      A        LDB    ,U+
02236A EB42 A1    C1      A CSWCK1 CMPA   ,U++
02237A EB44 25    07   EB4D        BLO    CSWCK8
02238A EB46 A1    5F      A        CMPA   -1,U
02239A EB48 23    06   EB50        BLS    CSWCK9
02240A EB4A 5A                     DECB
02241A EB4B 26    F5   EB42        BNE    CSWCK1
02242A EB4D 1C    FE          CSWCK8 CLC
02243A EB4F 39                     RTS
02244A EB50 1A    01          CSWCK9 SEC
02245A EB52 39                     RTS
02246                       *
02247                       ***************
02248                       *
```

```
02249                      *  LEFT
02250                      *
02251                      *******
02252                      *
02253             EB53    A CSLW
02254A EB53 0F    A3      A
02255A EB55 BD    EA0F    A
02256A EB58 BD    E3C7    A
02257A EB5B 8D    0B   EB68 CSLW10
02258A EB5D 24    FC   EB5B
02259                      *
02260A EB5F 8D    07   EB68 CSLW20
02261A EB61 25    FC   EB5F
02262                      *
02263A EB63 31    22      A
02264A EB65 7E    EA2E    A CSLW90
02265                      *
02266             EB68    A CSLWS
02267A EB68 0D    A3      A
02268A EB6A 26    0F   EB7B
02269A EB6C C6    80      A
02270A EB6E E5    A4      A
02271A EB70 27    02   EB74
02272A EB72 0C    A3      A
02273A EB74 8D    C5   EB3B CSLWS5
02274A EB76 30    1F      A
02275A EB78 31    3F      A
02276A EB7A 39
02277A EB7B 32    62      A CSLWS9
02278A EB7D 31    21      A
02279A EB7F 20    E4   EB65
02280                      *
02281A EB81       04      A CSWCKT
02282A EB82       30      A
     A EB83       39      A
02283A EB84       41      A
     A EB85       5A      A
02284A EB86       61      A
     A EB87       7A      A
02285A EB88       80      A
     A EB89       FF      A
02286                      *
02287                      *******
02288                      *
02289                      *  BACK
02290                      *
02291                      *******
02292                      *
02293A EB8A 39             GETBS9
02294             EB8B    A GETBS
02295A EB8B 9E    52      A
02296A EB8D A6    84      A
02297A EB8F 2B    F9   EB8A
02298A EB91 BD    EA0F    A
02299A EB94 BD    E3C7    A
02300                      *
02301                      *
02302                      *******
02303                      *
02304                      *  DEL
02305                      *
02306                      *******
02307                      *
02308             EB97    A GETDEL
02309A EB97 DC    4E      A
02310A EB99 34    06      A
02311A EB9B 9E    50      A
02312A EB9D 109E  52      A
02313A EBA0 E6    A4      A
02314A EBA2 C5    10      A
02315A EBA4 27    05   EBAB
02316A EBA6
     A EBA6 B5    D403    A
02317A EBA9
     A EBA9 35    86      A
02318             EBAB    A GETD10
02319A EBAB E6    21      A
02320A EBAD C5    80      A
02321A EBAF 26    25   EBD6
02322A EBB1 A6    01      A
02323A EBB3 A7    84      A
02324A EBB5 6D    A4      A
02325A EBB7 2A    02   EBBB
02326A EBB9 CA    80      A
02327A EBBB E7    A0      A GETD12
02328A EBBD 109C  47      A
02329A EBC0 26    07   EBC9
02330A EBC2 34    06      A
02331A EBC4 BD    E76B    A
02332A EBC7 35    06      A
02333             EBC9    A GETD15
02334A EBC9 BD    E51C    A
02335A EBCC BD    E9F3    A
02336A EBCF BD    E3C7    A
02337A EBD2 A6    84      A
02338A EBD4 26    D5   EBAB
02339
02340A EBD6 4F             GETD20
02341A EBD7 E6    A4      A
02342A EBD9 CA    40      A
02343A EBDB E7    A4      A
02344A EBDD A7    84      A
02345A EBDF BD    E51C    A
02346A EBE2 35    06      A
02347A EBE4 DD    4E      A
02348A EBE6 39
02349
```

```
 WORD   *
        *
********

EQU     *
CLR     GTFLD
JSR     CSL
JSR     CBADR
BSR     CSLWS
BCC     CSLW10   OFF

BSR     CSLWS
BCS     CSLW20   ON

LEAY    2,Y
JMP     GETENA

EQU     *
TST     GTFLD
BNE     CSLWS9
LDB     #AMFL
BITB    ,Y
BEQ     CSLWS5
INC     GTFLD
BSR     CSWCK
LEAX    -1,X
LEAY    -1,Y
RTS
LEAS    2,S        1 NEST REMOVE
LEAY    1,Y
BRA     CSLW90

FCB     4
FCB     '0,'9

FCB     'A,'Z

FCB     $61,$7A

FCB     $80,$FF

*****************
                *
SPACE & DELETE  *
                *
*****************

RTS
EQU     *
LDX     .CAAD
LDA     ,X
BMI     GETBS9   FIELD
JSR     CSL      BACK
JSR     CBADR
                  DELETE

**
 *
 *
 *
**

EQU     *
LDD     .CY
PSHS    D
LDX     .CBAD
LDY     .CAAD
LDB     ,Y
BITB    #AMPT
BEQ     GETD10
BUZZON
BITA    BUZZER
RTSPUL  D
PULS    D,PC
EQU     *
LDB     1,Y
BITB    #AMFL
BNE     GETD20
LDA     1,X
STA     ,X
TST     ,Y
BPL     GETD12
ORB     #AMFL      FIELD MARK
STB     ,Y+
CMPY    .BAAD
BNE     GETD15
PSHS    D
JSR     CTOB
PULS    D
EQU     *
JSR     CDSPC
JSR     CSR
JSR     CBADR
LDA     ,X
BNE     GETD10

CLRA
LDB     ,Y
ORB     #AMMD
STB     ,Y
STA     ,X
JSR     CDSPC
PULS    D
STD     .CY
RTS
```

```
02350                                   **************
02351                                   *            *
02352                                   *   CANCEL   *
02353                                   *            *
02354                                   **************
02355
02356               EBE7       A GETCAN EQU    *
02357A EBE7 86      01         A        LDA    #1
02358A EBE9                             SKIP1
       A EBE9       21         A        FCB    $21
02359
02360                                   **************
02361                                   *            *
02362                                   *   ENTER    *
02363                                   *            *
02364                                   **************
02365
02366               EBEA       A GETCR  EQU    *
02367A EBEA 4F                          CLRA
02368               EBEB       A GETCR1 EQU    *
02369A EBEB 97      32         A        STA    .EKEY
02370A EBED 32      62         A        LEAS   2,S      REMOVE RETURN
02371                                   *
02372                                   * SEND MODIFIED FIELD
02373                                   *
02374               EBEF       A GETC   EQU    *
02375A EBEF 96      32         A        LDA    .EKEY
02376A EBF1 B7      D383       A        STA    .COMAR+3
02377A EBF4 BD      E92F       A        JSR    GETRNG
02378A EBF7 109E    9F         A        LDY    GTTOPA
02379A EBFA C6      40         A        LDB    #AMMD
02380A EBFC A6      A4         A GETC05 LDA    ,Y
02381A EBFE 2A      02    EC02          BPL    GETC07
02382A EC00 33      A4         A        LEAU   ,Y
02383A EC02 E5      A0         A GETC07 BITB   ,Y+
02384A EC04 26      09    EC0F          BNE    GETC10
02385A EC06 109C    A1         A        CMPY   GTBTMA
02386A EC09 25      F1    EBFC          BLO    GETC05
02387                                   *
02388                                   * NO DATA
02389                                   *
02390A EC0B 7F      D384       A        CLR    .COMAR+4
02391A EC0E 39                 A        RTS
02392                                   *
02393               EC0F       A GETC10 EQU    *
02394A EC0F 31      3F         A        LEAY   -1,Y     SET THE FIRST OF MODIFY.
02395A EC11 E6      A4         A        LDB    ,Y
02396A EC13 C4      BF         A GETC15 ANDB   #$FF-AMMD
02397A EC15 E7      A0         A        STB    ,Y+      CLEAR THE MODIFYED DATA TAG.
02398A EC17 E6      A4         A        LDB    ,Y
02399A EC19 2A      F8    EC13          BPL    GETC15   MORE DATA IN FIELD.
02400A EC1B 31      C4         A        LEAY   ,U
02401A EC1D 30      C9 F830    A        LEAX   ..CRB-..ATB,U
02402A EC21 34      30         A        PSHS   X,Y
02403A EC23 BD      EA2B       A        JSR    GETENC
02404A EC26 35      30         A        PULS   X,Y
02405A EC28 CE      D384       A        LDU    #.COMAR+4
02406A EC2B 86      03         A        LDA    #3
02407A EC2D A7      C0         A        STA    ,U+
02408A EC2F 86      12         A        LDA    #$12     SET FIELD
02409A EC31 A7      C0         A        STA    ,U+
02410A EC33 DC      4E         A        LDD    .CY
02411A EC35 1E      89         A        EXG    A,B
02412A EC37 ED      C1         A        STD    ,U++
02413A EC39 31      21         A        LEAY   1,Y      PRE INC THE ATTR PTR.
02414                                   *
02415               EC3B       A GETC20 EQU    *
02416A EC3B A6      80         A        LDA    ,X+
02417A EC3D 27      14    EC53          BEQ    GETC35
02418A EC3F 1183    D400       A        CMPU   #.COMAR+128
02419A EC43 25      09    EC4E          BLO    GETC30
02420A EC45 BD      E0EE       A        JSR    CONTO
02421A EC48 7F      D384       A        CLR    .COMAR+4
02422A EC4B CE      D385       A        LDU    #.COMAR+5
02423               EC4E       A GETC30 EQU    *
02424A EC4E A7      C0         A        STA    ,U+
02425A EC50 7C      D384       A        INC    .COMAR+4
02426A EC53 E6      A0         A GETC35 LDB    ,Y+
02427A EC55 2A      E4    EC3B          BPL    GETC20
02428A EC57 39                 A        RTS
02429                                   *
02430                                   *
02431                                   * CONTROL KEY JUMP TABLE
02432                                   *
02433A EC58         E943       A GETJMP FDB    GETNOP   $00
02434A EC5A         E943       A        FDB    GETNOP   $01
02435A EC5C         EB53       A        FDB    CSLW     $02
02436A EC5E         EBE7       A        FDB    GETCAN   $03
02437A EC60         E943       A        FDB    GETNOP   $04
02438A EC62         EA9B       A        FDB    ELKEY    $05
02439A EC64         EB05       A        FDB    CSRW     $06
02440A EC66         E68C       A        FDB    GETBEL   $07
02441A EC68         EB8B       A        FDB    GETBS    $08
02442A EC6A         EADA       A        FDB    TAB      $09
02443A EC6C         E943       A        FDB    GETNOP   $0A
02444A EC6E         EA43       A        FDB    CSHOME   $0B
02445A EC70         EA81       A        FDB    GETCLR   $0C
02446A EC72         EBEA       A        FDB    GETCR    $0D
02447A EC74         E943       A        FDB    GETNOP   $0E
02448A EC76         E943       A        FDB    GETNOP   $0F
02449A EC78         E943       A        FDB    GETNOP   $10
02450A EC7A         EAD0       A        FDB    GETDUP   $11
02451A EC7C         EABF       A        FDB    INSERT   $12
02452A EC7E         E943       A        FDB    GETNOP   $13
02453A EC80         E943       A        FDB    GETNOP   $14
02454A EC82         E943       A        FDB    GETNOP   $15
02455A EC84         E943       A        FDB    GETNOP   $16
```

```
02456A ECB6       E943     A        FDB     GETNOP  $17
02457A ECB8       EBE7     A        FDB     GETCAN  $18
02458A ECBA       EA67     A        FDB     GETBFD  $19
02459A ECBC       EA5F     A        FDB     GETFFD  $1A
02460A ECBE       E943     A        FDB     GETNOP  $1B
02461A ECC0       E9C3     A GETJCS FDB     CSRL    $1C
02462A ECC2       E9CE     A        FDB     CSLL    $1D
02463A ECC4       E9D8     A        FDB     CSUL    $1E
02464A ECC6       E9E6     A        FDB     CSDL    $1F
02465                      *
02466                               TTL     GBADR - CONSOLE CODE -
02467                      ************************
02468                      *                      *
02469                      *  GET BUFFER ADDRESS  *
02470                      *                      *
02471                      ************************
02472                      *
02473             EC98     A GETBAD EQU     *
02474A EC98 DC    43       A        LDD     .BPY
02475A EC9A 1E    89       A        EXG     A,B
02476A EC9C FD   D383      A        STD     .COMAR+3
02477A EC9F 39                      RTS
02478                      *
02480                      *************************
02481                      *                       *
02482                      *  GET CHARACTER BLOCK  *
02483                      *                       *
02484                      *************************
02485                      *
02486A ECA0                         SORG    .WORK        .
                  D096     A .$ORG  SET     .WORK
02487A ECA0                         DS      CBCMD,1
                  D096     A CBCMD  EQU     .$ORG
                  D097     A .$ORG  SET     .$ORG+1
02488A ECA0                         DS      CBY1,1
                  D097     A CBY1   EQU     .$ORG
                  D098     A .$ORG  SET     .$ORG+1
02489A ECA0                         DS      CBX1,1
                  D098     A CBX1   EQU     .$ORG
                  D099     A .$ORG  SET     .$ORG+1
02490A ECA0                         DS      CBY2,1
                  D099     A CBY2   EQU     .$ORG
                  D09A     A .$ORG  SET     .$ORG+1
02491A ECA0                         DS      CBX2,1
                  D09A     A CBX2   EQU     .$ORG
                  D09B     A .$ORG  SET     .$ORG+1
02492A ECA0                         DS      CBXTOP,1
                  D09B     A CBXTOP EQU     .$ORG
                  D09C     A .$ORG  SET     .$ORG+1
02493A ECA0                         DS      CPBAT,1
                  D09C     A CPBAT  EQU     .$ORG
                  D09D     A .$ORG  SET     .$ORG+1
02494A ECA0                         DS      CPBCNT,1
                  D09D     A CPBCNT EQU     .$ORG
                  D09E     A .$ORG  SET     .$ORG+1
02495A ECA0                         DS      CBSAVE,2
                  D09E     A CBSAVE EQU     .$ORG
                  D0A0     A .$ORG  SET     .$ORG+2
02496                      *
02497A ECA0                         FLGENT  CGBLK1,CGBLK2,CBCMD
     A ECA0       86       A CGBLK2 FCB     $86     LDA #
     A ECA1 4F               CGBLK1 CLRA
     A ECA2 97    96       A        STA     CBCMD
02498A ECA4 8D    29   ECCF         BSR     CBINIT
02499A ECA6 CE   D383      A        LDU     #CMDPRM
02500A ECA9 6F    C0       A        CLR     ,U+
02501A ECAB 8D    53   ED00 CGB100  BSR     CBLADR
02502A ECAD 96    96       A        LDA     CBCMD
02503A ECAF 27    04   ECB5         BEQ     CGB120
02504A ECB1 A6    A4       A        LDA     ,Y
02505A ECB3 8D    06   ECBB         BSR     CGBDT
02506A ECB5 A6    84       A CGB120 LDA     ,X
02507A ECB7 8D    02   ECBB         BSR     CGBDT
02508A ECB9 20    F0   ECAB         BRA     CGB100
02509                      *
02510             ECBB     A CGBDT  EQU     *
02511A ECBB 1183 D400      A        CMPU    #COMEND
02512A ECBF 25    08   ECC9         BLO     CGBDT5
02513A ECC1 BD   E0EE      A        JSR     CONTO
02514A ECC4 CE   D383      A        LDU     #CMDPRM
02515A ECC7 6F    C0       A        CLR     ,U+
02516A ECC9 7C   D383      A CGBDT5 INC     CMDPRM
02517A ECCC A7    C0       A        STA     ,U+
02518A ECCE 39                      RTS
02519                      *
02520             ECCF     A CBINIT EQU     *
02521A ECCF FC   D383      A        LDD     CMDPRM
02522A ECD2 BD   E3D9      A        JSR     CAXCR
02523A ECD5 24    0A   ECE1         BCC     CBIN05
02524A ECD7 97    98       A        STA     CBX1
02525A ECD9 D7    97       A        STB     CBY1
02526A ECDB FC   D385      A        LDD     CMDPRM+2
02527A ECDE BD   E3D9      A        JSR     CAXCR
02528A ECE1 1024 F3E6 E0CB CBIN05   LBCC    ERR61
02529A ECE5 91    98       A        CMPA    CBX1
02530A ECE7 24    05   ECEE         BHS     CBIN50
02531A ECE9 97    98       A        STA     CBX1
02532A ECEB B6   D383      A        LDA     CMDPRM
02533A ECEE 97    9A       A CBIN50 STA     CBX2
02534A ECF0 D1    97       A        CMPB    CBY1
02535A ECF2 24    05   ECF9         BHS     CBIN60
02536A ECF4 D7    97       A        STB     CBY1
02537A ECF6 F6   D384      A        LDB     CMDPRM+1
02538A ECF9 D7    99       A CBIN60 STB     CBY2
02539A ECFB D6    98       A        LDB     CBX1
02540A ECFD D7    9B       A        STB     CBXTOP
02541A ECFF 39                      RTS
02542                      *

02543             ED00     A CBLADR EQU
02544A ED00 DC    97       A        LDD
02545A ED02 DD    9E       A        STD
02546A ED04 91    99       A        CMPA
02547A ED06 22    10   ED18         BHI
02548A ED08 BD   E211      A        JSR
02549A ED0B DC    97       A        LDD
02550A ED0D 5C                      INCB
02551A ED0E D1    9A       A        CMPB
02552A ED10 23    03   ED15         BLS
02553A ED12 4C                      INCA
02554A ED13 D6    9B       A        LDB
02555A ED15 DD    97       A CBLAD7 STD
02556A ED17 39                      RTS
02557A ED18 32    62       A CBLEND LEAS
02558A ED1A 7E   E282      A        JMP
02559                      *
02560                      **************
02561                      *
02562                      *  PUT CHARACT
02563                      *
02564                      **************
02565                      *
02566A ED1D                         FLGENT
     A ED1D       86       A CPBLK2 FCB
     A ED1E 4F               CPBLK1 CLRA
     A ED1F 97    96       A        STA
02567A ED21 8D    AC   ECCF         BSR
02568A ED23 CE   D387      A        LDU
02569A ED26 EC    C1       A        LDD
02570A ED28 DD    9C       A        STD
02571A ED2A 8D    D4   ED00 CPB100  BSR
02572A ED2C 96    96       A        LDA
02573A ED2E 26    03   ED33         BNE
02574A ED30 96    9C       A        LDA
02575A ED32                         SKIP2
     A ED32       8C       A        FCB
02576A ED33 8D    1C   ED51 CPB120  BSR
02577A ED35 1F    89       A        TFR
02578A ED37 8D    18   ED51         BSR
02579A ED39 A7    84       A        STA
02580A ED3B E7    A4       A        STB
02581A ED3D 34    06       A        PSHS
02582A ED3F DC    9E       A        LDD
02583A ED41 BD   E22A      A        JSR
02584A ED44 AE    E4       A        LDX
02585A ED46 EF    E4       A        STU
02586A ED48 1E    01       A        EXG
02587A ED4A BD   E521      A        JSR
02588A ED4D 35    40       A        PULS
02589A ED4F 20    D9   ED2A         BRA
02590                      *
02591             ED51     A CPBDT  EQU
02592A ED51 0A    9D       A        DEC
02593A ED53 2A    10   ED65         BPL
02594A ED55 96    02       A        LDA
02595A ED57 2A    BF   ED18         BPL
02596A ED59 BD   E11A      A        JSR
02597A ED5C CE   D383      A        LDU
02598A ED5F A6    C0       A        LDA
02599A ED61 97    9D       A        STA
02600A ED63 20    EC  .ED51         BRA
02601A ED65 A6    C0       A CPBDT5 LDA
02602A ED67 39                      RTS
02603                      *
02605                      *
02606                      *  TABSET
02607                      *
02608             ED68     A TABSET EQU
02609A ED68 8E   D383      A        LDX
02610A ED6B 108E D022      A        LDY
02611A ED6F C6    0A       A        LDB
02612A ED71 7E   E44E      A        JMP
02614                      *
02615                      *  CONSOLE CON
02616                      *
02617             ED74     A CNSCTL EQU
02618A ED74 B6   D383      A        LDA
02619A ED77 97    1F       A CNSINT STA
02620A ED79 85    01       A        BITA
02621A ED7B 27    12   ED8F         BEQ
02622A ED7D 8D    15   ED94         BSR
02623A ED7F 96    1A       A CURON  LDA
02624A ED81 44                      LSRA
02625A ED82 25    06   ED8A         BCS
02626A ED84 86    06       A        LDA
02627A ED86 97    1B       A        STA
02628A ED88 0F    1C       A        CLR
02629A ED8A 86    01       A CURON5 LDA
02630A ED8C 97    1A       A        STA
02631A ED8E 39                      RTS
02632A ED8F 8D    0C   ED9D CUROFF  BSR
02633A ED91 0F    1A       A        CLR
02634A ED93 39                      RTS
02635A ED94 8D    1B   EDB1 CURSET  BSR
02636A ED96 DC    43       A        LDD
02637A ED98 DD    4E       A        STD
02638A ED9A 7E   E3C7      A        JMP
02639A ED9D 8D    12   EDB1 CURSCL  BSR
02640A ED9F 96    1C       A CURCLR LDA
02641A EDA1 27    0D   EDB0         BEQ
02642A EDA3 A6    9F D050  A        LDA
02643A EDA7 E6    9F D052  A        LDB
02644A EDAB BD   E51C      A        JSR
02645A EDAE 0F    1C       A        CLR
02646A EDB0 39               CURRTS RTS
02647A EDB1 96    1A       A CURSTP LDA
02648A EDB3 8A    80       A        ORA
```

```
*
CBY1
CBSAVE
CBY2
CBLEND
BADR
CBY1

CBX2
CBLAD7

CBXTOP
CBY1

2,S     REMOVE
ASSFLD  WINDOW TOP FIELD MADE & RETURN

************
           *
ER BLOCK   *
           *
************

CPBLK1,CPBLK2,CBCMD
$86     LDA #

CBCMD
CBINIT
#CMDPRM+4
,U++
CPBAT
CBLADR
CBCMD
CPB120
CPBAT

$8C     CMPX #
CPBDT
A,B
CPBDT
,X
,Y
D
CBSAVE
BVAD
,S
,S
D,X
CDSP
U
CPB100

*
CPBCNT
CPBDT5
..CONT
CBLEND
CONTI
#CMDPRM
,U+
CPBCNT
CPBDT
,U+




*
#CMDPRM TABSET DATA
#.TABST TABSET AREA
#10     80 BITS FOR USE
MOVE    SET & RETURN TO MAIN

TROL

*
CMDPRM
.CSCTL  SET THE CONTROL WORD.
#CSLCUR DO YOU WANT CURSOR?
CUROFF  NO, GO OFF THE CURSOR.
CURSET  YES, SET THE CURSOR ADDRESS.
.CURSR  WAS THE CURSOR SELECTED?

CURON5  YES.
#$06    NO. SET THE CURSOR TIMER.
.CURTM
.CURST  CLEAR THE CURSOR STATUS.
#$01    SET THE CURSOR SELECT BIT.
.CURSR

CURSCL  STOP AND CLEAR THE CURSOR.
.CURSR  CLEAR THE CURSOR SELECT BIT.

CURSTP  STOP THE CURSOR
.BPY    SET THE CURRENT POSITION.
.CY
CBADR
CURSTP  STOP THE CURSOR.
.CURST  IS IT REVERCE?
CURRTS  NO.
[.CBAD] YES.
[.CAAD]
CDSPC   DISPLAY NORMAL CHAR.
.CURST  CLEAR THE REVERSE FLAG.

.CURSR  STOP THE CURSOR.
#$80
```

```
02649A EDB5 20    04   EDBB         BRA    CURSTA
02650A EDB7 96    1A      A CURGO   LDA    .CURSR   CLEAR THE CURSOR STOP.
02651A EDB9 84    7F      A         ANDA   #$7F
02652A EDBB 97    1A      A CURSTA  STA    .CURSR
02653A EDBD 39                      RTS

02655             0001    A CSLCUR  EQU    $01
02656             0002    A CSLORD  EQU    $02
02657             0004    A CSLTAB  EQU    $04
02658             0008    A CSLPAG  EQU    $08
02659             0010    A CSLSUS  EQU    $10
02660             0020    A CSLALF  EQU    $20
02662             EDBE    A INDICT  EQU    *
02663                       * PROGRAMABLE FUNCTION KEY'S STRING
02664                       * INDICATOR.
02665                       *
02666A EDBE                         SORG   .BOXX1
                  D05F    A .$ORG   SET    .BOXX1
02667A EDBE                         DS     INDLP1,1
                  D05F    A INDLP1  EQU    .$ORG
                  D060    A .$ORG   SET    .$ORG+1
02668A EDBE                         DS     INDLP2,1
                  D060    A INDLP2  EQU    .$ORG
                  D061    A .$ORG   SET    .$ORG+1
02669A EDBE                         DS     INDLP3,1
                  D061    A INDLP3  EQU    .$ORG
                  D062    A .$ORG   SET    .$ORG+1
02670A EDBE                         DS     INDPF,1
                  D062    A INDPF   EQU    .$ORG
                  D063    A .$ORG   SET    .$ORG+1
02671A EDBE                         DS     INDBSV,1
                  D063    A INDBSV  EQU    .$ORG
                  D064    A .$ORG   SET    .$ORG+1
02672                       *
02673A EDBE 96    36      A         LDA    ..FS
02674A EDC0 27    20   EDE2         BEQ    INDIC9   NO DISPLAY DIRECT
02675A EDC2 86    0F      A         LDA    #15      NUMBER OF CHAR FOR DISPLAY
02676A EDC4 D6    2F      A         LDB    ..CWD    IS THE CHARACTER SIZE SMALL?
02677A EDC6 27    02   EDCA         BEQ    INDIC2   YES.
02678A EDC8 86    07      A         LDA    #7       NO.
02679A EDCA 97    62      A INDIC2  STA    INDPF
02680A EDCC 96    2D      A         LDA    ..BCL
02681A EDCE 97    63      A         STA    INDBSV   BCL SAVE
02682A EDD0 0F    2D      A         CLR    ..BCL    BACK BLACK
02683A EDD2 8E    D2C0    A         LDX    #.PFTBL
02684A EDD5 96    41      A         LDA    ..DMAP+6
02685A EDD7 4C                      INCA
02686A EDD8 8D    09   EDE3         BSR    INDI
02687A EDDA 96    41      A         LDA    ..DMAP+6
02688A EDDC 8D    05   EDE3         BSR    INDI
02689A EDDE D6    63      A         LDB    INDBSV
02690A EDE0 D7    2D      A         STB    ..BCL
02691A EDE2 39              INDIC9  RTS
02692                       *
02693             EDE3    A INDI    EQU    *
02694A EDE3 BD    E226    A         JSR    BVADL    VRAM ADDR. U
02695A EDE6 1F    03      A         TFR    D,U
02696A EDE8 C6    05      A         LDB    #5
02697A EDEA D7    5F      A         STB    INDLP1
02698A EDEC 8D    16   EE04 INDI3   BSR    PFDSP
02699A EDEE 0A    5F      A         DEC    INDLP1
02700A EDF0 26    FA   EDEC         BNE    INDI3
02701A EDF2 39                      RTS
02702                       *
02703A EDF3 4F              INDSPC  CLRA
02704A EDF4 5F                      CLRB
02705                       *
02706             EDF5    A INDCHR  EQU    *
02707A EDF5 34    56      A         PSHS   A,B,X,U
02708A EDF7 30    C4      A         LEAX   ,U
02709A EDF9 BD    E521    A         JSR    CDSP
02710A EDFC EC    64      A         LDD    4,S
02711A EDFE D3    37      A         ADDD   ..VCU    VRAM NEXT
02712A EE00 ED    64      A         STD    4,S
02713A EE02                         RTSPUL (A,B,X,U)
     A EE02 35    D6      A         PULS   A,B,X,U,PC
02714                       *
02715             EE04    A PFDSP   EQU    *
02716A EE04 34    10      A         PSHS   X        PF PTR SAVE
02717A EE06 E6    80      A         LDB    ,X+      INTERRUPT
02718A EE08 2B    03   EE0D         BMI    PFDSP3   YES.
02719A EE0A C6    0D      A         LDB    #$0D     NO.
02720A EE0C                         SKIP2
     A EE0C       8C      A         FCB    $8C      CMPX #
02721A EE0D C6    0B      A PFDSP3  LDB    #$0B
02722A EE0F 96    62      A         LDA    INDPF    DISPLAY CHR
02723A EE11 97    61      A         STA    INDLP3   COUNT
02724A EE13 A6    80      A PFDSP5  LDA    ,X+
02725A EE15 8D    DE   EDF5         BSR    INDCHR
02726A EE17 0A    61      A         DEC    INDLP3
02727A EE19 26    F8   EE13         BNE    PFDSP5
02728A EE1B 35    10      A         PULS   X
02729A EE1D 30    88 10   A         LEAX   PFSZ,X   NEXT PF
02730A EE20 20    D1   EDF3         BRA    INDSPC
02731                               TTL    GRAPHIC CODE - GRAPH -
02733             EE22    A CHRLN   EQU    *
02734A EE22 CE    D380    A         LDU    #.COMAR
02735A EE25 A6    46      A         LDA    6,U
02736A EE27 E6    48      A         LDB    8,U
02737A EE29 BD    E3D9    A         JSR    CAXCR
02738A EE2C 24    07   EE35         BCC    CHRLER
02739A EE2E A6    4A      A         LDA    10,U
02740A EE30 E6    4C      A         LDB    12,U
02741A EE32 BD    E3D9    A         JSR    CAXCR
02742A EE35 1024  F292 E0CB CHRLER  LBCC   ERR61
02743A EE39 6F    45      A         CLR    5,U
02744A EE3B 6F    47      A         CLR    7,U
```

```
A EE3D 6F   49        A        CLR    9,U
A EE3F 6F   4B        A        CLR    11,U
A EE41 86   FF        A        LDA    #$FF
A EE43 B7   D096      A        STA    LNCHR
A EE46 20   0E   EE56          BRA    LINEDO
                        *
                        * LINE COMMAND
                        *
                        *    .COMAR+3  COLOR CODE
                        *    .COMAR+4  FUNCTION CODE
                        *    .COMAR+5  X CO-ORDINATE OF P1
                        *    .COMAR+7  Y CO-ORDINATE OF P1
                        *    .COMAR+9  X CO-ORDINATE OF P2
                        *    .COMAR+11 Y CO-ORDINATE OF P2
                        *    .COMAR+13 BOX FLAG
                        *
                        *
            EE48      A LINE   EQU    *
A EE48 FC   D383      A        LDD    .COMAR+3
A EE4B 84   07        A        ANDA   #$07
A EE4D C1   05        A        CMPB   #5
A EE4F 24   23   EE74          BHS    LINE90  FUNCTION CODE ERROR
A EE51 DD   56        A        STD    .GCL    SET COLOR AND FUNCTION CODE.
A EE53 BD   E456      A        JSR    INITLN
A EE56 CE   D385      A LINEDO LDU    #CMDPRM+2
A EE59 37   30        A        PULU   X,Y
A EE5B BD   E446      A        JSR    GAXCHK
A EE5E 9F   67        A        STX    .LX1
A EE60 109F 69        A        STY    .LY1
A EE63 37   30        A        PULU   X,Y
A EE65 BD   E446      A        JSR    GAXCHK
A EE68 B6   D38D      A        LDA    .COMAR+13
A EE6B 26   0A   EE77          BNE    LINE10
                        *
                        *  DRAW A STRAIGHT LINE
                        *
A EE6D 9F   6B        A        STX    .LX2
A EE6F 109F 6D        A        STY    .LY2
A EE72 20   56   EECA          BRA    LINES
                        *
A EE74 7E   E0D4      A LINE90 JMP    ERR64
                        *
A EE77 4A               LINE10 DECA
A EE78 26   2D   EEA7          BNE    LINE20
                        *
                        *  DRAW A RECTANGLE
                        *
A EE7A 9F   6B        A        STX    .LX2
A EE7C BE   D387      A        LDX    .COMAR+7
A EE7F 9F   6D        A        STX    .LY2
A EE81 8D   47   EECA          BSR    LINES   X1,Y1-X2,Y1
A EE83 BE   D385      A        LDX    .COMAR+5
A EE86 9F   6B        A        STX    .LX2
A EE88 BE   D38B      A        LDX    .COMAR+11
A EE8B 9F   6D        A        STX    .LY2
A EE8D 8D   3B   EECA          BSR    LINES   X1,Y1-X1,Y2
A EE8F BE   D389      A        LDX    .COMAR+9
A EE92 9F   67        A        STX    .LX1
A EE94 BE   D38B      A        LDX    .COMAR+11
A EE97 9F   69        A        STX    .LY1
A EE99 8D   2F   EECA          BSR    LINES   X2,Y2-X1,Y2
A EE9B BE   D389      A        LDX    .COMAR+9
A EE9E 9F   6B        A        STX    .LX2
A EEA0 BE   D387      A        LDX    .COMAR+7
A EEA3 9F   6D        A        STX    .LY2
A EEA5 20   23   EECA          BRA    LINES   X2,Y2-X2,Y1
                        *
            EEA7      A LINE20 EQU    *
                        *
                        *  PAINT A RECTANGLE
                        *
A EEA7 9F   6B        A        STX    .LX2
A EEA9 109F 6D        A        STY    .LY2
A EEAC 96   6A        A        LDA    .LY1+1
A EEAE D6   6E        A        LDB    .LY2+1
A EEB0 D0   6A        A        SUBB   .LY1+1
A EEB2 24   03   EEB7          BHS    LINE30
A EEB4 50                      NEGB
A EEB5 96   6E        A        LDA    .LY2+1
A EEB7 97   6A        A LINE30 STA    .LY1+1
A EEB9 97   6E        A        STA    .LY2+1
A EEBB 5C                      INCB
A EEBC 34   04        A        PSHS   B
A EEBE 8D   0A   EECA LINE35   BSR    LINES
A EEC0 0C   6A        A        INC    .LY1+1
A EEC2 0C   6E        A        INC    .LY2+1
A EEC4 6A   E4        A        DEC    ,S
A EEC6 26   F6   EEBE          BNE    LINE35
A EEC8                         RTSPUL B
A EEC8 35   84        A        PULS   B,PC
                        *
                        *
                        *  END OF "LINE" COMMAND
                        *
                        *
                        *    "LINES"
                        *
                        *    DRAW A STRAIGHT LINE
                        *
                        *    INPUT
                        *
                        *     .LX1 X CO-ORDINATE OF POINT1
                        *     .LY1 Y CO-ORDINATE OF POINT1
                        *     .LX2 X CO-ORDINATE OF POINT2
                        *     .LY2 Y CO-ORDINATE OF POINT2
                        *
            D096      A LNCHR  EQU    .WORK
            D097      A LNBPY  EQU    LNCHR+1
            D099      A LNDX   EQU
            D09B      A LNDY   EQU
            D09D      A LNUX   EQU
            D09F      A LNUY   EQU
            D0A1      A LNPY   EQU
            D0A3      A LNLP   EQU
            D0A5      A LINKLN EQU
                        *
            EECA      A LINES  EQU
A EECA 8E   0001      A        LDX
A EECD CE   FFFF      A        LDU
A EED0 DC   6B        A        LDD
A EED2 93   67        A        SUBD
A EED4 2B   04   EEDA          BMI
A EED6 9F   9D        A        STX
A EED8 20   06   EEE0          BRA
A EEDA DF   9D        A LINES1 STU
A EEDC DC   67        A        LDD
A EEDE 93   6B        A        SUBD
A EEE0 C3   0001      A LINES2 ADDD
A EEE3 DD   99        A        STD
A EEE5 DD   A3        A        STD
A EEE7 DC   6D        A        LDD
A EEE9 93   69        A        SUBD
A EEEB 2B   04   EEF1          BMI
A EEED 9F   9F        A        STX
A EEEF 20   06   EEF7          BRA
A EEF1 DF   9F        A LINES3 STU
A EEF3 DC   69        A        LDD
A EEF5 93   6D        A        SUBD
A EEF7 C3   0001      A LINES4 ADDD
A EEFA DD   9B        A        STD
                        *
A EEFC 96   96        A        LDA
A EEFE 26   0E   EF0E          BNE
A EF00 9E   99        A        LDX
A EF02 30   1F        A        LEAX
A EF04 27   74   EF7A          BEQ
A EF06 9E   9B        A        LDX
A EF08 30   1F        A        LEAX
A EF0A 1027 0088 EF96          LBEQ
                        *
A EF0E CC   0000      A LNCDSP LDD
A EF11 DD   A1        A        STD
A EF13 DC   67        A        LDD
A EF15 DD   5A        A        STD
A EF17 DC   69        A        LDD
A EF19 DD   58        A        STD
                        *
            EF1B      A LINES5 EQU
A EF1B BD   EF50      A        JSR
A EF1E DC   A1        A        LDD
A EF20 D3   9B        A        ADDD
A EF22 DD   A1        A        STD
                        *
A EF24 93   99        A LINES6 SUBD
A EF26 23   0F   EF37          BLS
A EF28 DD   A1        A        STD
A EF2A D6   59        A        LDB
A EF2C DB   A0        A        ADDB
A EF2E D7   59        A        STB
                        * LDD LNPY
                        * SUBD LNDX
                        * BLO LINES8
A EF30 BD   EF50      A        JSR
A EF33 DC   A1        A        LDD
A EF35 20   ED   EF24          BRA
                        *
            EF37      A LINES7 EQU
A EF37 26   08   EF41          BNE
A EF39 DD   A1        A        STD
A EF3B D6   59        A        LDB
A EF3D DB   A0        A        ADDB
A EF3F D7   59        A        STB
A EF41 DC   5A        A LINES8 LDD
A EF43 D3   9D        A        ADDD
A EF45 DD   5A        A        STD
A EF47 9E   A3        A        LDX
A EF49 30   1F        A        LEAX
A EF4B 9F   A3        A        STX
A EF4D 26   CC   EF1B          BNE
A EF4F 39               LINES9 RTS
                        *
A EF50 96   96        A LNPOIN LDA
A EF52 27   23   EF77          BEQ
A EF54 D6   5B        A        LDB
A EF56 96   59        A        LDA
A EF58 34   06        A        PSHS
A EF5A BD   E211      A        JSR
A EF5D 35   06        A        PULS
A EF5F BD   E22A      A        JSR
A EF62 34   06        A        PSHS
A EF64 B6   D384      A        LDA
A EF67 A7   84        A        STA
A EF69 E6   A4        A        LDB
A EF6B C4   80        A        ANDB
A EF6D FA   D383      A        ORB
A EF70 E7   A4        A        STB
A EF72 35   10        A        PULS
A EF74 7E   E521      A        JMP
A EF77 7E   E493      A LNDOT  JMP
                        * DRAW A LINE
                        *
A EF7A 9E   67        A LINEY  LDX
A EF7C 109E 69        A        LDY
A EF7F 96   9F        A        LDA
A EF81 2A   03   EF86          BPL
```

```
LNBPY+2
LNDX+2
LNDY+2
LNUX+2
LNUY+2
LNPY+2
LNLP+2

*
#1      1 UNIT
#-1     -1 UNIT
.LX2
.LX1    DX=X2-X1;
LINES1  IF DX>=0
LNUX    THEN UX=1;
LINES2
LNUX    ELSE DO; UX=-1;
.LX1    DX=X1-X2;
.LX2    END;
#1      DX=DX+1;
LNDX
LNLP
.LY2
.LY1    DY=Y2-Y1;
LINES3  IF DY>=0
LNUY    THEN UY=1;
LINES4
LNUY    ELSE DO;UY=-1;
.LY1    DY=Y1-Y2;
.LY2    END;
#1      DY=DY+1;
LNDY
LNCHR   SHALL I DISPLAY CHARACTER?
LNCDSP  YES.
LNDX
-1,X
LINEY
LNDY
-1,X
LINEX

#0
LNPY    PY=0;
.LX1
.GX     GX=X1;
.LY1
.GY     GY=Y1;

*
LNPOIN  CALL GPDSP;
LNPY
LNDY
LNPY    PY=PY+DY;

LNDX    DO WHILE(PY>DX);
LINES7
LNPY
.GY+1
LNUY+1
.GY+1   GY=GY+UY;



LNPOIN  CALL GPDSP;
LNPY
LINES6  END;

*
LINES8  PY .NE. DX THEN GOTO LINES8
LNPY    PY=0;
.GY+1   GY=GY+UY
LNUY+1
.GY+1
.GX
LNUX
.GX     GX=GX+UX;
LNLP
-1,X
LNLP
LINES5

LNCHR
LNDOT
.GX+1
.GY+1
D
BADR    81/7/7
D
BVAD
D
.COMAR+4
,X
,Y
#$80
.COMAR+3
,Y
X
CDSP
GPDSP
THAT IS PARALLEL TO Y AXIS.

.LX1
.LY1    ASSUME POINT-1 COORDINATES.
LNUY    IS IT A NEGATIVE DIRECTION?
LINEY2  NO.
```

```
A EF83 109E 6D          A        LDY   .LY2     YES, THE START POINT IS THE POINT2.
A EF86 BD   E3E0        A LINEY2 JSR   GADDR    GET THE VRAM ADDRESS OF START POINT.
A EF89 30   C4          A LINEY4 LEAX  ,U       SAVE IT TO X.
A EF8B BD   E49B        A        JSR   GPDSPR   DISPLAY ADOT.
A EF8E 33   88 50       A        LEAU  GXBSZ,X
A EF91 0A   9C          A        DEC   LNDY+1
A EF93 26   F4    EF89           BNE   LINEY4
A EF95 39                 LINEXR RTS
                          *
                          * DRAW A LINE THAT IS PARRALLEL TO X AXIS.
                          *
A EF96 9E   67          A LINEX  LDX   .LX1
A EF98 DE   6B          A        LDU   .LX2
A EF9A 96   9D          A        LDA   LNUX     IS IT NEGATIVE?
A EF9C 2A   02    EFA0           BPL   LINEX2   NO.
A EF9E 1E   13          A        EXG   X,U      YES EXCHANG LX1 AND LX2
A EFA0 9F   9D          A LINEX2 STX   LNUX     SET LEFT X TO LNUX.(DESTROY UNIT OF STEP)
A EFA2 DF   9F          A        STU   LNUY     SET RIGHT X TO LNUY.(DESTROY UNIT OF STEP)
A EFA4 96   9E          A        LDA   LNUX+1
A EFA6 84   07          A        ANDA  #$07     GET LEFT BIT POS
A EFA8 8E   E396        A        LDX   #BAS
A EFAB E6   86          A        LDB   A,X
A EFAD 58                        LSLB
A EFAE 5A                        DECB           LEFT SIDE BIT MASK
A EFAF 96   A0          A        LDA   LNUY+1
A EFB1 84   07          A        ANDA  #$07     GET RIGHT BIT POS
A EFB3 A6   86          A        LDA   A,X
A EFB5 4A                        DECA
A EFB6 43                        COMA           RIGHT SIDE BIT MASK
A EFB7 34   06          A        PSHS  A,B      SAVE THE SIDE MASK.
A EFB9 DC   9D          A        LDD   LNUX
A EFBB BD   E3FF        A        JSR   LSRD3
A EFBE 34   04          A        PSHS  B
A EFC0 DC   9F          A        LDD   LNUY
A EFC2 BD   E3FF        A        JSR   LSRD3
A EFC5 E0   E0          A        SUBB  ,S+      GET THE NUMBER OF BYTE
A EFC7 26   07    EFD0           BNE   LINEX3   INTERVAL IS GRATER THAN 1.
A EFC9 E6   E0          A        LDB   ,S+      ONLY ONNE BYTE.
A EFCB E4   E0          A        ANDB  ,S+      LOGICAL AND THE BOTH SIDE BIT MASK.
A EFCD 7E   F00A        A        JMP   LNRSID   SET BIT IMG. AND RETURN

A EFD0 D7   A3          A LINEX3 STB   LNLP     SAVE
A EFD2 E6   E0          A        LDB   ,S+
A EFD4 C1   FF          A        CMPB  #$FF     IS THE RIGHT SIDE FULL BYTE?
A EFD6 27   05    EFDD           BEQ   LINEX4   YES.
A EFD8 BD   F00A        A        JSR   LNRSID   NO, PUT BYTE WITH MASK.
A EFDB 0A   A3          A        DEC   LNLP
A EFDD E6   E0          A LINEX4 LDB   ,S+      IS THE LEFT SIDE FULL BYTE?
A EFDF 2B   0D    EFEE           BMI   LINEX5   YES.
A EFE1 BD   F00D        A        JSR   LNLSID   NO, PUT BYTE WITH MASK.
A EFE4 9E   9D          A        LDX   LNUX     LEFT SIDE IS POINTED NEXT.
A EFE6 30   08          A        LEAX  8,X
A EFE8 9F   9D          A        STX   LNUX
A EFEA 0A   A3          A        DEC   LNLP
A EFEC 2B   A7    EF95           BMI   LINEXR   NO BYTE REMAIN. GO FINISH.
    EFEE                A LINEX5 EQU   *
A EFEE 109E 69          A        LDY   .LY1     LOAD THE Y COORDINATE.
A EFF1 9E   9D          A        LDX   LNUX     LOAD THE LEFT X.
A EFF3 BD   E3E0        A        JSR   GADDR    GET LOWEST DOT ADDRESS.
A EFF6 D6   A3          A        LDB   LNLP
A EFF8 9D   82          A        JSR   BLULIN
A EFFA 33   C9 4000 A            LEAU  ..VR,U
A EFFE D6   A3          A        LDB   LNLP
A F000 9D   85          A        JSR   REDLIN
A F002 33   C9 4000 A            LEAU  ..VG-..VR,U
A F006 D6   A3          A        LDB   LNLP
A F008 0E   88          A        JMP   GRELIN
                          *
A F00A 9E   9F          A LNRSID LDX   LNUY     LOAD THE RIGHT X COORDINATE
A F00C                           SKIP2
  A F00C    8C          A        FCB   $8C      CMPX #
A F00D 9E   9D          A LNLSID LDX   LNUX     LOAD THE LEFT X COORDINATE.
A F00F 109E 69          A        LDY   .LY1     LOAD THE Y COORDINATE.
A F012 34   04          A        PSHS  B        SAVE BIT MASK.
A F014 BD   E3E0        A        JSR   GADDR    GET VRAM ADDR.
A F017 30   C4          A        LEAX  ,U       TFR U TO X.
A F019 31   E4          A        LEAY  ,S       Y POINT TO BIT MASK
A F01B 5F                        CLRB
A F01C 9D   6F          A        JSR   BLUSID   BLUE
A F01E 30   C9 4000 A            LEAX  ..VR,U   X PTR RED VRAM
A F022 5F                        CLRB
A F023 9D   72          A        JSR   REDSID   RED
A F025 30   C9 8000 A            LEAX  ..VG,U   X PTR GREEN VRAM
A F029 5F                        CLRB
A F02A 9D   75          A        JSR   GRESID   GREEN
A F02C                           RTSPUL B
  A F02C 35 84          A        PULS  B,PC



A F02E      F612        A LNXFNC FDB   LNXOR,LNXAND PSET
  A F030    F628        A
A F032      F612        A        FDB   LNXOR,LNXAND PRESET
  A F034    F628        A
A F036      F612        A        FDB   LNXOR,LNXNOP OR
  A F038    F638        A
A F03A      F638        A        FDB   LNXNOP,LNXAND AND
  A F03C    F628        A
A F03E      F63F        A        FDB   LNXEOR,LNXNOP EOR
  A F040    F638        A


                          *
A F042 4F                 LNAND  CLRA
A F043                           SKIP2
```

```
      A F043 8C      A       FCB   $8C     CMPX #
03056A F044 86  FF      A LNOR   LDA   #$FF
03057A F046                      VACON
      A F046 B5  D409   A        BITA  VACC
03058A F049 12                   NOP
03059A F04A A7  C5      A LNORDO STA   B,U
03060A F04C 5A                   DECB
03061A F04D 2A  FB   F04A        BPL   LNORDO
03062A F04F                      VACOFF
      A F04F B7  D409   A        STA   VACC
03063A F052 39            LNNOP  RTS
03064                     *
03065A F053               LNEOR  VACON
      A F053 B5  D409   A        BITA  VACC
03066A F056 12                   NOP
03067A F057 63  C5      A LNEORD COM   B,U
03068A F059 5A                   DECB
03069A F05A 2A  FB   F057        BPL   LNEORD
03070A F05C                      VACOFF
      A F05C B7  D409   A        STA   VACC
03071A F05F 39                   RTS
03072                     *
03073A F060      F044   A LNFUNC FDB   LNOR,LNAND
      A F062     F042   A
03074A F064      F044   A        FDB   LNOR,LNAND
      A F066     F042   A
03075A F068      F044   A        FDB   LNOR,LNNOP
      A F06A     F052   A
03076A F06C      F052   A        FDB   LNNOP,LNAND
      A F06E     F042   A
03077A F070      F053   A        FDB   LNEOR,LNNOP
      A F072     F052   A
03078                     *
03079                     *** END OF SUBROUTINE "LINES"
03081                     *
03082                     *
03083                     * CHAIN COMMAND
03084                     *
03085                     *   .COMAR+3  COLOR CODE
03086                     *   .COMAR+4  FUNCTION CODE
03087                     *   .COMAR+5  NUMBER OF POINT
03088                     *   .COMAR+6  CHAINED POINTS
03089                     *
03090            F074   A CHAIN  EQU   *
03091A F074 FC  D383    A        LDD   CMDPRM  A=COLOR,B=FUNCTION
03092A F077 C1  05      A        CMPB  #5
03093A F079 24  32   F0AD        BHS   CHAIN7  FUNCTION CODE ERROR
03094A F07B DD  56      A        STD   .GCL
03095A F07D BD  E456    A        JSR   INITLN
03096A F080 CE  D385    A        LDU   #CMDPRM+2
03097A F083 A6  C0      A        LDA   ,U+
03098A F085 81  01      A        CMPA  #1
03099A F087 23  27   F0B0        BLS   CHAIN8  NUMBER OF POINTS ERROR
03100A F089 81  1E      A        CMPA  #30
03101A F08B 22  23   F0B0        BHI   CHAIN8  NUMBER OF POINTS ERROR
03102A F08D 4A                   DECA
03103A F08E 37  30      A        PULU  X,Y
03104A F090 BD  E446    A        JSR   GAXCHK
03105A F093 9F  67      A CHAIN2 STX   .LX1
03106A F095 109F 69     A        STY   .LY1
03107A F098 37  30      A        PULU  X,Y
03108A F09A BD  E446    A        JSR   GAXCHK
03109A F09D 9F  6B      A        STX   .LX2
03110A F09F 109F 6D     A        STY   .LY2
03111A F0A2 34  72      A        PSHS  X,Y,U,A
03112A F0A4 BD  EECA    A        JSR   LINES
03113A F0A7 35  72      A        PULS  X,Y,U,A
03114A F0A9 4A                   DECA
03115A F0AA 26  E7   F093        BNE   CHAIN2
03116A F0AC 39                   RTS
03117A F0AD 7E  E0D4    A CHAIN7 JMP   ERR64
03118                     *
03119A F0AD 7E  E0D4    A CHAIN7 JMP   ERR64
03120A F0B0 7E  E0D7    A CHAIN8 JMP   ERR65
03121                     *
03122                     *  END OF "CHAIN" COMMAND
03123                     *
03125                     *
03126                     *
03127                     * SYMBOL COMMAND
03128                     *   .COMAR+3  COLOR CODE
03129                     *   .COMAR+4  FUNCTION CODE
03130                     *   .COMAR+5  ANGLE CODE
03131                     *   .COMAR+6  MUL OF WIDTH
03132                     *   .COMAR+7  MUL OF HEIGHT
03133                     *   .COMAR+8  X COORDINATES
03134                     *   .COMAR+10 Y COORDINATES
03135                     *   .COMAR+12 NUMBER OF CHARACTER
03136                     *   .COMAR+13 CHARACTER STRING
03137                     *
03138                     *
03139            D096   A WAX    EQU   .WORK    PTR OF WK W AX
03140            D098   A HAX    EQU   .WORK+2  PTR OF WK H AX
03141            D09A   A WSV    EQU   .WORK+4  W DOT STEP
03142            D09C   A HSV    EQU   .WORK+6  H DOT STEP
03143            D09E   A WCSV   EQU   .WORK+8  W CHR DOT STEP
03144            D0A0   A HCSV   EQU   .WORK+10 H CHR DOT STEP
03145            D0A2   A NOTF   EQU   .WORK+12 FONT NOT FLAG
03146            D0A3   A CPTR   EQU   .WORK+13 CHARACTER PTR
03147            D0A5   A WOP    EQU   .WORK+15 W ORG
03148            D0A7   A HOP    EQU   .WORK+17 H ORG
03149            D0A9   A FMRK   EQU   .WORK+19 FONT TEST MASK
03150            D0AA   A FPTR   EQU   .WORK+20 FONT PTR
03151            D0AC   A LW     EQU   .WORK+22 LOOP W
03152            D0AD   A LH     EQU   .WORK+23 LOOP H
03153            D0AE   A SYMFNT EQU   .WORK+24 TEMP FONT AREA
03154                     *
```

```
03155                          *
03156            F0B3        A SYMBOL EQU
03157A F0B3 FC   D383        A        LDD
03158A F0B6 0F   A2          A        CLR
03159A F0B8 C1   05          A        CMPB
03160A F0BA 22   F1   F0AD            BHI
03161A F0BC 26   03   F0C1            BNE
03162A F0BE 03   A2          A        COM
03163A F0C0 5F                        CLRB
03164A F0C1 BD   E46F        A SYMBO0 JSR
03165A F0C4 B6   D385        A        LDA
03166A F0C7 44                        LSRA
03167A F0C8 CC   D05A        A        LDD
03168A F0CB 8E   D058        A        LDX
03169A F0CE 108E D388        A        LDY
03170A F0D2 CE   D38A        A        LDU
03171A F0D5 24   04   F0DB            BCC
03172A F0D7 1E   01          A        EXG
03173A F0D9 1E   23          A        EXG
03174A F0DB 34   76          A SYMBO1 PSHS
03175                          *
03176                          * 0,S COORD
03177                          * 2,S COORD
03178                          * 4,S ORIGI
03179                          * 6,S ORIGI
03180                          *
03181A F0DD CE   D385        A        LDU
03182A F0E0 4F                        CLRA
03183A F0E1 E6   41          A        LDB
03184A F0E3 27   04   F0E9            BEQ
03185A F0E5 DD   9E          A        STD
03186A F0E7 E6   42          A        LDB
03187A F0E9 1027 EFF6 E0E3 SYMBO4     LBEQ
03188A F0ED DD   A0          A        STD
03189A F0EF E6   C4          A        LDB
03190A F0F1 C4   03          A        ANDB
03191A F0F3 C1   02          A        CMPB
03192A F0F5 25   02   F0F9            BLO
03193A F0F7 C8   01          A        EORB
03194A F0F9 34   04          A        PSHS
03195A F0FB 5F                        CLRB
03196A F0FC 64   E4          A        LSR
03197A F0FE 24   06   F106            BCC
03198A F100 93   9E          A        SUBD
03199A F102 DD   9E          A        STD
03200A F104 C6   FE          A        LDB
03201A F106 5C                 SYMBO5 INCB
03202A F107 DD   9A          A        STD
03203A F109 4F                        CLRA
03204A F10A 5F                        CLRB
03205A F10B 64   E0          A        LSR
03206A F10D 24   06   F115            BCC
03207A F10F 93   A0          A        SUBD
03208A F111 DD   A0          A        STD
03209A F113 C6   FE          A        LDB
03210A F115 5C                 SYMBO6 INCB
03211A F116 DD   9C          A        STD
03212A F118 8E   D38D        A        LDX
03213A F11B 9F   A3          A        STX
03214            F11D        A SYMB10 EQU
03215A F11D B6   D38C        A        LDA
03216A F120 26   02   F124            BNE
03217A F122                           RTSPUL
      A F122 35  F6          A        PULS
03218A F124 A6   9F D0A3     A SYMB13 LDA
03219A F128 BD   E3BB        A        JSR
03220A F12B 108E D0AE        A        LDY
03221A F12F C6   08          A        LDB
03222A F131 A6   80          A SYMB15 LDA
03223A F133 0D   A2          A        TST
03224A F135 27   01   F138            BEQ
03225A F137 43                        COMA
03226A F138 A7   A0          A        STA
03227A F13A 5A                        DECB
03228A F13B 26   F4   F131            BNE
03229A F13D FC   D388        A        LDD
03230A F140 DD   5A          A        STD
03231A F142 FC   D38A        A        LDD
03232A F145 DD   58          A        STD
03233A F147 BD   E387        A        JSR
03234A F14A 25   02   F14E            BCS
03235A F14C                           RTSPUL
      A F14C 35  F6          A        PULS
03236A F14E 86   80          A SYMB17 LDA
03237A F150 97   A9          A        STA
03238            F152        A SYMB20 EQU
03239A F152 AE   F8 06       A        LDX
03240A F155 AF   F8 02       A        STX
03241A F158 BE   D0AE        A        LDX
03242            F15B        A SYMB30 EQU
03243A F15B 9F   AA          A        STX
03244A F15D BD   E387        A        JSR
03245A F160 24   08   F16A            BCC
03246A F162 96   A9          A        LDA
03247A F164 A5   9F D0AA     A        BITA
03248A F168 26   0A   F174            BNE
03249            F16A        A SYMB33 EQU
03250A F16A DC   A0          A        LDD
03251A F16C AE   62          A        LDX
03252A F16E E3   84          A        ADDD
03253A F170 ED   84          A        STD
03254A F172 20   34   F1A8            BRA
03255                          *
03256            F174        A SYMB35 EQU
03257A F174 B6   D387        A        LDA
03258A F177 97   AD          A        STA
03259            F179        A SYMB40 EQU
```

```
*        ENTRY
CMDPRM
NOTF
#5       NOT?
CHAIN7
SYMB00   NO.
NOTF
         FUNCTION IS PSET
GPDSIR
.COMAR+5 ANGLE

#.GX
#.GY
#.COMAR+8
#.COMAR+10
SYMB01
D,X
Y,U
U,Y,X,D
INATES PTR OF WIDTH AXIS
INATES PTR OF HEIGHT AXIS
NAL COORDINATES PTR OF WIDTH AXIS
NAL COORDINATES PTR OF HEIGHT AXIS

#.COMAR+5

1,U      MUL OF WIDTH
SYMB04
WCSV
2,U      MUL OF HEIGHT
ERR69    PARM ERROR
HCSV
,U       ANGLE CODE
#$03
#2
*+4
#$01
B

,S
SYMB05
WCSV
WCSV
#$FE

WSV



,S+
SYMB06
HCSV
HCSV
#$FE

HSV
#.COMAR+13
CPTR     VHAR PTR
*        START DISPLAY
.COMAR+12 CHAR END?
SYMB13   NO, GO DO MORE.
(U,Y,X,D)
U,Y,X,D,PC
[CPTR]   CHR LOAD
CFADR    FONT ORIGINAL
#SYMFNT  FONT STORE ADDR
#CFH     NUMBER OF FONT BYTE
0,X+
NOTF
*+3

         NOT FUNCTION
0,Y+     STORE

SYMB15   REMAIN
.COMAR+8
.GX
.COMAR+10
.GY
GAXC     CHECK AXIS
SYMB17
(U,Y,X,D)
U,Y,X,D,PC
#$80
FMRK     TEST MARK INIT
*
[6,S]
[2,S]
#SYMFNT  FONT ADDR INIT
*
FPTR     SET FONT PTR
GAXC
SYMB33
FMRK
[FPTR]   DOT ON?
SYMB35   ON
*
HCSV     CLEAR
2,S
,X
,X
SYMB60

*
.COMAR+7
LH       HEIGHT DOT
*
```

```
03260A F179 B6    D386     A         LDA    .COMAR+6 WIDTH DOT
03261A F17C 97    AC       A         STA    LW
03262             F17E     A SYMB50  EQU    *
03263A F17E BD    E493     A         JSR    GPDSP    ***DISPLAY***
03264A F181 AE    E4       A         LDX    ,S
03265A F183 EC    84       A         LDD    ,X
03266A F185 D3    9A       A         ADDD   WSV
03267A F187 ED    84       A         STD    ,X
03268A F189 BD    E387     A         JSR    GAXC
03269A F18C 24    04   F192          BCC    SYMB55   OVER
03270A F18E 0A    AC       A         DEC    LW
03271A F190 26    EC   F17E          BNE    SYMB50
03272             F192     A SYMB55  EQU    *
03273A F192 EC    F8 04    A         LDD    [4,S]
03274A F195 ED    F4       A         STD    [,S]
03275A F197 AE    62       A         LDX    2,S
03276A F199 EC    84       A         LDD    ,X
03277A F19B D3    9C       A         ADDD   HSV
03278A F19D ED    84       A         STD    ,X
03279A F19F BD    E387     A         JSR    GAXC
03280A F1A2 24    0D   F1B1          BCC    SYMB70
03281A F1A4 0A    AD       A         DEC    LH
03282A F1A6 26    D1   F179          BNE    SYMB40
03283             F1A8     A SYMB60  EQU    *
03284A F1A8 9E    AA       A         LDX    FPTR
03285A F1AA 30    01       A         LEAX   1,X      PTR NEXT FONT DOT
03286A F1AC BC    D0B6     A         CMPX   #SYMFNT+CFH END ?
03287A F1AF 26    AA   F15B          BNE    SYMB30   NO
03288             F1B1     A SYMB70  EQU    *
03289A F1B1 AE    64       A         LDX    4,S
03290A F1B3 EC    84       A         LDD    ,X
03291A F1B5 D3    9E       A         ADDD   WCSV
03292A F1B7 ED    84       A         STD    ,X
03293A F1B9 ED    F4       A         STD    [,S]
03294A F1BB 04    A9       A         LSR    FMRK     SHIFT RIGHT MARK
03295A F1BD 26    93   F152          BNE    SYMB20
03296A F1BF 7A    D3BC     A         DEC    .COMAR+12 ONE CHR END
03297A F1C2 9E    A3       A         LDX    CPTR     PTR NEXT CHR
03298A F1C4 30    01       A         LEAX   1,X
03299A F1C6 9F    A3       A         STX    CPTR
03300A F1C8 7E    F11D     A         JMP    SYMB10
03301                              *
03302                              ***********************************
03303                              *
03305                              *
03306                              *     CHANGE  COMMAND
03307                              *
03308                              *    .COMAR+3  X1
03309                              *    .COMAR+5  Y1
03310                              *    .COMAR+7  X2
03311                              *    .COMAR+9  Y2
03312                              *    .COMAR+11 NUMBER OF COLOR
03313                              *    .COMAR+12 COLOR
03314                              *
03315                              *
03316             D096     A CHGCL   EQU    .WORK
03317             D09E     A CHGI    EQU    CHGCL+8
03318             D09F     A CHGJ    EQU    CHGI+1
03319             D0A0     A CHGK    EQU    CHGJ+1
03320                              *
03321             F1CB     A CHANGE  EQU    *
03322A F1CB BD    E406     A         JSR    BOXINV
03323A F1CE D7    A0       A         STB    CHGK
03324A F1D0 8E    D096     A         LDX    #CHGCL
03325A F1D3 86    07       A         LDA    #7
03326A F1D5 A7    86       A CHANG0  STA    A,X
03327A F1D7 4A                       DECA
03328A F1D8 2A    FB   F1D5          BPL    CHANG0   CHANGE TABLE INIT VALUE
03329A F1DA F6    D38B     A         LDB    .COMAR+11
03330A F1DD 27    51   F230          BEQ    CHANGX
03331A F1DF C1    08       A         CMPB   #8
03332A F1E1 22    4D   F230          BHI    CHANGX
03333A F1E3 108E  D38C     A         LDY    #.COMAR+12
03334A F1E7 D7    9E       A         STB    CHGI
03335A F1E9 EC    A1       A CHANG1  LDD    0,Y++
03336A F1EB 84    07       A         ANDA   #7
03337A F1ED C4    07       A         ANDB   #7
03338A F1EF E7    86       A         STB    A,X
03339A F1F1 0A    9E       A         DEC    CHGI
03340A F1F3 26    F4   F1E9          BNE    CHANG1   CHANGE TABLE MADE
03341                              *
03342A F1F5 0F    57       A         CLR    .GF      FUNCTION IS "SET"
03343             F1F7     A CHANG4  EQU    *
03344A F1F7 9E    5F       A         LDX    .BOXX1
03345A F1F9 109E  61       A         LDY    .BOXY1
03346A F1FC 96    A0       A         LDA    CHGK
03347A F1FE 4C                       INCA
03348A F1FF 97    9F       A         STA    CHGJ
03349A F201 BD    E3E0     A         JSR    GADDR
03350A F204 8E    D096     A         LDX    #CHGCL
03351             F207     A CHANG5  EQU    *
03352A F207 BD    E39E     A         JSR    GPGETR
03353A F20A A6    86       A         LDA    A,X
03354A F20C 97    56       A         STA    .GCL
03355A F20E 34    54       A         PSHS   X,U,B
03356A F210 BD    E46D     A         JSR    GPDSPI
03357A F213 35    54       A         PULS   X,U,B
03358A F215 34    54       A         PSHS   X,U,B
03359A F217 BD    E49B     A         JSR    GPDSPR
03360A F21A 35    54       A         PULS   X,U,B
03361A F21C 33    C8 50    A         LEAU   GXBSZ,U
03362A F21F 0A    9F       A         DEC    CHGJ
03363A F221 26    E4   F207          BNE    CHANG5
03364                              *
03365A F223 9E    5F       A         LDX    .BOXX1
03366A F225 9C    63       A         CMPX   .BOXX2
03367A F227 27    06   F22F          BEQ    CHANG9
```

```
03368A F229 30   01        A        LEAX   1,X
03369A F22B 9F   5F        A        STX    .BOXX1
03370A F22D 20   C8   F1F7          BRA    CHANG4
03371                             *
03372A F22F 39                    CHANG9 RTS
03373A F230 7E   E0DD      A CHANGX JMP    ERR67
03374                             *
03375                             *
03376                             ***** CHANGE COMMAND END
03378                             *   GET BLOCK1 COMMAND
03379                             *
03380                             *   .COMAR+3  X1
03381                             *   .COMAR+5  Y1
03382                             *   .COMAR+7  X2
03383                             *   .COMAR+9  Y2
03384                             *   .COMAR+11 NUMBER OF COLOR
03385                             *   .COMAR+12 COLOR CODES
03386                             *
03387                             *
03388            D096      A GBADX  EQU    .WORK    TOP ADDRESS OF X.
03389            D098      A GBDPX  EQU    GBADX+2 BIT NO. OF X
03390            D099      A GBNBX  EQU    GBDPX+1 BYTES,ON DATA
03391            D09A      A GBNX   EQU    GBNBX+1 DOTS,X
03392            D09C      A GBNY   EQU    GBNX+2  DOTS,Y
03393            D09D      A GBI    EQU    GBNY+1   TEMP
03394            D09E      A GBJ    EQU    GBI+1    TEMP
03395            D09F      A GBCL   EQU    GBJ+1    24 MODIFING CONSTANT
03396                             *             FOR COLOR
03397            D0B7      A GBNCL  EQU    GBCL+24 NUMBER OF COLOR
03398            D0B8      A GBPIX  EQU    GBNCL+1 PUT INDEX
03399            D0BA      A GBDIX  EQU    GBPIX+2 DATA INDEX
03400            D0BC      A GBIX   EQU    GBDIX+2 BIT NO,SHIFTED TOP
03401            D0BD      A GBTMIX EQU    GBIX+1
03402            D0BF      A GBK    EQU    GBTMIX+2
03403            D0C0      A GBVB   EQU    GBK+1    BLUE DATA
03404            D0C1      A GBVR   EQU    GBVB+1   RED DATA
03405            D0C2      A GBVG   EQU    GBVR+1   GREEN DATA
03406            D0C3      A GBSV   EQU    GBVG+1   SHIFT VALUE
03407            D0C4      A GBDATA EQU    GBSV+1   83 BYTES,PLASTIC AREA.
03408                             *
03409A F233 7E   E0DD      A GBLK19 JMP    ERR67
03410                             *
03411            F236      A GBLK1  EQU    *
03412A F236 BD   F2C0      A        JSR    GBINT
03413A F239 B6   D38B      A        LDA    .COMAR+11 NO. OF COLOR
03414A F23C 27   F5   F233          BEQ    GBLK19   CHECK
03415A F23E 81   08        A        CMPA   #8
03416A F240 22   F1   F233          BHI    GBLK19
03417A F242 97   B7        A        STA    GBNCL
03418A F244 8E   D38C      A        LDX    #.COMAR+12 PTR,COLOR CODE
03419A F247 CE   D09F      A        LDU    #GBCL    COLOR TABLE
03420            F24A      A GBLK11 EQU    *
03421A F24A BD   F2E2      A        JSR    GBSCL    BLUE BIT
03422A F24D BD   F2E2      A        JSR    GBSCL    RED BIT
03423A F250 BD   F2E2      A        JSR    GBSCL    GREEN BIT
03424A F253 30   01        A        LEAX   1,X
03425A F255 7A   D38B      A        DEC    .COMAR+11
03426A F258 26   F0   F24A          BNE    GBLK11
03427                             *
03428                             * - OUTPUT DATA -
03429                             *   .COMAR+3 NUMBER OF BYTE
03430                             *   .COMAR+4  PATTERN DATA WHEN AFTER THE FIRST
03431                             *
03432A F25A BD   F3DD      A        JSR    GBOINT
03433            F25D      A GBLK12 EQU    *
03434A F25D 96   99        A        LDA    GBNBX
03435A F25F 97   9D        A        STA    GBI      LOOP COUNTER
03436A F261 108E D0C6      A        LDY    #GBDATA+2
03437A F265 9E   BA        A        LDX    GBDIX
03438A F267                  GBLK13 VACON
     A F267 B5   D409      A        BITA   VACC
03439A F26A 12                      NOP
03440A F26B E6   B4        A        LDB    ..VB,X
03441A F26D D7   C0        A        STB    GBVB
03442A F26F E6   B9 4000 A          LDB    ..VR,X
03443A F273 D7   C1        A        STB    GBVR
03444A F275 E6   B9 8000 A          LDB    ..VG,X
03445A F279 30   01        A        LEAX   1,X
03446A F27B D7   C2        A        STB    GBVG
03447A F27D                         VACOFF
     A F27D B7   D409      A        STA    VACC
03448A F280 6F   A4        A        CLR    0,Y
03449A F282 CE   D09F      A        LDU    #GBCL
03450A F285 96   B7        A        LDA    GBNCL    LOOP COUNTER
03451A F287 E6   C0        A GBLK14 LDB    0,U+
03452A F289 D8   C0        A        EORB   GBVB     EOR BLUE
03453A F28B E7   21        A        STB    1,Y      TEMP,SAVE
03454A F28D E6   C0        A        LDB    0,U+
03455A F28F D8   C1        A        EORB   GBVR
03456A F291 EA   21        A        ORB    1,Y
03457A F293 E7   21        A        STB    1,Y
03458A F295 E6   C0        A        LDB    0,U+
03459A F297 D8   C2        A        EORB   GBVG
03460A F299 EA   21        A        ORB    1,Y
03461A F29B 53                      COMB
03462A F29C EA   A4        A        ORB    0,Y
03463A F29E E7   A4        A        STB    0,Y      SET
03464A F2A0 4A                      DECA
03465A F2A1 26   E4   F287          BNE    GBLK14
03466A F2A3 31   21        A        LEAY   1,Y
03467A F2A5 0A   9D        A        DEC    GBI
03468A F2A7 26   BE   F267          BNE    GBLK13
03469                             *
03470                             *   DOT PICKUP END
03471                             *
03472A F2A9 BD   F306      A        JSR    GBPLA    SHIFT AND PUT
03473                             *
```

```
03474A F2AC 9E   BA        A        LDX
03475A F2AE 30   88 50     A        LEAX
03476A F2B1 9F   BA        A        STX
03477A F2B3 0A   9C        A        DEC
03478A F2B5 26   A6   F25D          BNE
03479A F2B7 0D   BC        A        TST
03480A F2B9 27   04   F2BF          BEQ
03481A F2BB 96   C4        A        LDA
03482A F2BD 8D   2A   F2E9          BSR
03483A F2BF 39                    GBLK17 RTS
03484                             *
03485                             *
03486                             *   "GBINT"
03487                             *
03488                             *   INITIAL
03489                             *   COMMAND
03490                             *
03491            F2C0      A GBINT  EQU
03492A F2C0 BD   E406      A        JSR
03493A F2C3 30   01        A        LEAX
03494A F2C5 9F   9A        A        STX
03495A F2C7 5C                      INCB
03496A F2C8 D7   9C        A        STB
03497A F2CA DC   5F        A        LDD
03498A F2CC BD   E3F6      A        JSR
03499A F2CF 9F   96        A        STX
03500A F2D1 D7   98        A        STB
03501A F2D3 DC   63        A        LDD
03502A F2D5 BD   E3F6      A        JSR
03503A F2D8 1F   10        A        TFR
03504A F2DA 93   96        A        SUBD
03505A F2DC 5C                      INCB
03506A F2DD D7   99        A        STB
03507A F2DF 0F   BC        A        CLR
03508A F2E1 39                      RTS
03509                             *
03510                             *
03511                             *
03512                             *   PICKUP
03513                             *   EXTEND
03514                             *   AND SET
03515                             *
03516            F2E2      A GBSCL  EQU
03517A F2E2 64   84        A        LSR
03518A F2E4 56                      RORB
03519A F2E5 1D                      SEX
03520A F2E6 A7   C0        A        STA
03521A F2E8 39                      RTS
03522                             *
03523                             *
03524                             *   SET ONE
03525                             *   IF AREA
03526                             *   BEFORE
03527                             *
03528                             *
03529            F2E9      A GBPUT  EQU
03530A F2E9 34   14        A        PSHS
03531A F2EB F6   D383      A        LDB
03532A F2EE 9E   B8        A        LDX
03533A F2F0 C1   7C        A        CMPB
03534A F2F2 25   07   F2FB          BLO
03535A F2F4 BD   E0EE      A        JSR
03536A F2F7 5F                      CLRB
03537A F2F8 8E   D384      A        LDX
03538            F2FB      A GBPUT1 EQU
03539A F2FB A7   80        A        STA
03540A F2FD 5C                      INCB
03541A F2FE F7   D383      A        STB
03542A F301 9F   B8        A        STX
03543A F303 35   14        A        PULS
03544A F305 39                      RTS
03545                             *
03546                             *
03547                             *
03548                             *   "GBPLA"
03549                             **
03550                             *   PLASTIC
03551                             *   AND SET
03552                             *
03553            F306      A GBPLA  EQU
03554                             * GBSV=GBIX-
03555A F306 96   BC        A        LDA
03556A F308 90   98        A        SUBA
03557A F30A 97   C3        A        STA
03558                             * IF GBSV>4
03559A F30C 81   04        A        CMPA
03560A F30E 2F   06   F316          BLE
03561A F310 80   08        A        SUBA
03562A F312 97   C3        A        STA
03563A F314 20   08   F31E          BRA
03564                             * IF GBSV<-4
03565A F316 81   FC        A GBPLA1 CMPA
03566A F318 2C   04   F31E          BGE
03567A F31A 8B   08        A        ADDA
03568A F31C 97   C3        A        STA
03569                             * NEXT:
03570            F31E      A GBPLA2 EQU
03571                             * IF GBSV<0
03572A F31E 0D   C3        A        TST
03573A F320 2B   17   F339          BMI
03574                             * DO GBI=1 T
03575A F322 96   C3        A        LDA
03576A F324 97   9D        A        STA
03577A F326 27   28   F350 GBPLA3 BEQ
03578                             * DO GBJ=1 T
03579A F328 96   99        A        LDA
03580A F32A 97   9E        A        STA
```

```
 GBDIX
 GXBSZ,X INC Y CO-ORDINATE
 GBDIX
 GBNY    DEC Y LOOP
 GBLK12
 GBIX
 GBLK17  REMAIN IS ZERO
 GBDATA
 GBPUT




IZE "GBLK1" OR "GBLK2"



 *
 BOXINV
 1,X
 GBNX

 GBNY
 .BOXX1
 GXOFF    ADDRESS,BIT NO.
 GBADX    X,ADDRESS
 GBDPX    X,BIT NO.
 .BOXX2
 GXOFF
 X,D
 GBADX

 GBNBX    NUMBER OF BYTE X
 GBIX




COLOR BIT,
COLOR BIT TO A BYTE
 EXTENDED BYTE TO TABLE.

 *
 O,X        PICKUP

            EXTEND
 ,U+        SET



 BYTE TO COMM. AREA
 IS FULL,DO CONINUOUS SEND
SETTING BYTE.



 *
 B,X
 .COMAR+3
 GBPIX
 #128-4
 GBPUT1
 CONTO    SEND

 #.COMAR+4
 *
 O,X+

 .COMAR+3
 GBPIX
 B,X







 OPERATION OF OUTPUT DATA
 DATA TO COM. AREA

 *
GBDPX; N PLACE OF SHIFT
 GBIX
 GBDPX
 GBSV
THEN DO; GBSV=GBSV-8;GOTO NEXT;END;
 #4
 GBPLA1
 #8
 GBSV
 GBPLA2
THEN    GBSV=GBSV+8;
 #-4
 GBPLA2
 #8
 GBSV
 *
THEN GOTO MSHIFT;
 GBSV
 GBPLA5
 GBSV;
 GBSV
 GBI
 GBPL10
 GBNB+1;
 GBNBX
 GBJ
```

```
03581A F32C 8E    DOC6    A        LDX     #GBDATA+2
03582                             *
03583             F32F    A GBPLA4 EQU     *
03584                             * T=SUBSTR(GBDATA(GBJ+2),8,1);
03585                             * GBDATA(GBJ+2)=C¥¥SUBSTR(GBDATA(GBJ+2),1,7);
03586                             * C=T;
03587A F32F 66    80      A        ROR     0,X+
03588A F331 0A    9E      A        DEC     GBJ
03589A F333 2A    FA    F32F       BPL     GBPLA4
03590                             * END ;
03591                             * END ; GOTO OUTPUT;
03592A F335 0A    9D      A        DEC     GBI
03593A F337 20    ED    F326       BRA     GBPLA3
03594                             * MSHIFT :
03595             F339    A GBPLA5 EQU     *
03596                             * DO GBI=-1 TO GSV STEP -1
03597A F339 96    C3      A        LDA     GBSV
03598A F33B 97    9D      A        STA     GBI
03599                             *
03600A F33D 8E    DOC6    A GBPLA6 LDX     #GBDATA+2
03601A F340 96    99      A        LDA     GBNBX
03602A F342 30    86      A        LEAX    A,X
03603                             * DO GBJ=1 TO GBNBX+1;
03604                             *
03605A F344 97    9E      A        STA     GBJ
03606             F346    A GBPLA7 EQU     *
03607                             * T=SUBSTR(GBDATA(GBNBX+2-GBJ),1,1);
03608                             * GBDATA(GBNBXX+2-GBJ)=SUBSTR(GBDATA(GBNB+2-GBJ),2,7)¥¥C;
03609                             * C=T;
03610A F346 69    82      A        ROL     0,-X
03611                             * END;
03612A F348 0A    9E      A        DEC     GBJ
03613A F34A 2A    FA    F346       BPL     GBPLA7
03614                             * END;
03615A F34C 0C    9D      A        INC     GBI
03616A F34E 26    ED    F33D       BNE     GBPLA6
03617                             * OUTPUT
03618             F350    A GBPL10 EQU     *
03619                             * P=ADDR(GBDATA(3));
03620A F350 8E    DOC6    A        LDX     #GBDATA+2
03621                             * IF GBSV+GBDPX ) 7 THEN P=P+1;
03622A F353 96    C3      A        LDA     GBSV
03623A F355 9B    98      A        ADDA    GBDPX
03624A F357 81    07      A        CMPA    #7
03625A F359 2F    02    F35D       BLE     GBPL11
03626A F35B 30    01      A        LEAX    1,X
03627             F35D    A GBPL11 EQU     *
03628                             * IF GBSV+GBDPX ( 0 THEN P=P-1;
03629A F35D 4D                     TSTA
03630A F35E 2A    02    F362       BPL     GBPL12
03631A F360 30    1F      A        LEAX    -1,X
03632             F362    A GBPL12 EQU     *
03633                             * MASK=BAS(GBIX+1)*2-1;
03634A F362 CE    E396    A        LDU     #BAS
03635A F365 96    BC      A        LDA     GBIX
03636A F367 E6    C6      A        LDB     A,U
03637A F369 58                     LSLB
03638A F36A 5A                     DECB
03639                             * P-)D=(P-)D&MASK)¥(GBDATA(1)&MASK);
03640A F36B 1F    98      A        TFR     B,A
03641A F36D A4    84      A        ANDA    0,X
03642A F36F A7    84      A        STA     0,X
03643A F371 53                     COMB
03644A F372 D4    C4      A        ANDB    GBDATA
03645A F374 EA    84      A        ORB     0,X
03646A F376 E7    84      A        STB     0,X
03647                             * N=GBNX+GBIX;
03648                             * GBIX=MOD(N,8);
03649                             * N=N/8;
03650A F378 DC    9A      A        LDD     GBNX
03651A F37A DB    BC      A        ADDB    GBIX
03652A F37C 89    00      A        ADCA    #0
03653A F37E 1F    12      A        TFR     X,Y     ASSIGN Y TO P
03654A F380 BD    E3F6    A        JSR     GXOFF
03655A F383 D7    BC      A        STB     GBIX
03656A F385 1F    10      A        TFR     X,D     ASSIGN B TO N
03657                             * DO I=1 TO N;
03658A F387 5D                     TSTB
03659A F388 27    08    F392 GBPL14 BEQ    GBPL15
03660                             * CALL GBPUT(-)D);P=P+1;
03661A F38A A6    A0      A        LDA     0,Y+
03662A F38C BD    F2E9    A        JSR     GBPUT
03663                             * END;
03664A F38F 5A                     DECB
03665A F390 20    F6    F388       BRA     GBPL14
03666             F392    A GBPL15 EQU     *
03667                             * GBDATA(1)=P-)D;
03668A F392 A6    A4      A        LDA     0,Y
03669A F394 97    C4      A        STA     GBDATA
03670A F396 39                     RTS
03671                             *
03672                             *
03673                             *
03674                             *
03675                             * GET BLOCK2 COMMAND
03676                             *     .COMAR+3  X1
03677                             *     .COMAR+5  Y1
03678                             *     .COMAR+7  X2
03679                             *     .COMAR+9  Y2
03680                             *
03681                             *
03682                             *
03683             F397    A GBLK2  EQU     *
03684A F397 BD    F2C0    A        JSR     GBINT   CO-ORDINATE DATA
03685A F39A 8D    41    F3DD       BSR     GBOINT  OUTPUT PROCESS
03686A F39C 9E    BA      A        LDX     GBDIX
03687A F39E 30    84      A        LEAX    ..VB,X
```

```
A F3A0 8D   0E    F3B0          BSR   GBLK2S
A F3A2 9E   BA       A          LDX   GBDIX
A F3A4 30   89 4000  A          LEAX  ..VR,X
A F3A8 8D   06    F3B0          BSR   GBLK2S
A F3AA 9E   BA       A          LDX   GBDIX
A F3AC 30   89 8000  A          LEAX  ..VG,X
                              *
            F3B0     A GBLK2S EQU   *
A F3B0 96   9C       A          LDA   GBNY
A F3B2 97   BF       A          STA   GBK     LOOP COUNTER SET
            F3B4     A GBLK25 EQU   *
A F3B4 9F   BD       A          STX   GBTMIX  SAVE VRAM ADDR
A F3B6 108E D0C6     A          LDY   #GBDATA+2
A F3BA D6   99       A          LDB   GBNBX
A F3BC                          VACON
  A F3BC B5 D409     A          BITA  VACC
A F3BF BD   E44E     A          JSR   MOVE     VRAM READ
A F3C2                          VACOFF
  A F3C2 B7 D409     A          STA   VACC
                              *
A F3C5 BD   F306     A          JSR   GBPLA    1 LINE OUT
                              *
A F3C8 9E   BD       A          LDX   GBTMIX
A F3CA 30   88 50    A          LEAX  GXBSZ,X NEXT LINE
A F3CD 0A   BF       A          DEC   GBK      DEC LOOP C
A F3CF 26   E3    F3B4          BNE   GBLK25
                              *
A F3D1 0D   BC       A          TST   GBIX     REMAIN ?
A F3D3 27   07    F3DC          BEQ   GBLK27
A F3D5 96   C4       A          LDA   GBDATA
A F3D7 BD   F2E9     A          JSR   GBPUT
A F3DA 0F   BC       A          CLR   GBIX     REMAIN IS 0
A F3DC 39                GBLK27 RTS
                              *
            F3DD     A GBOINT EQU   *
A F3DD 7F   D383     A          CLR   .COMAR+3
A F3E0 8E   D384     A          LDX   #.COMAR+4
A F3E3 9F   B8       A          STX   GBPIX
A F3E5 96   62       A          LDA   .BOXY1+1 Y CO-ORDINATE
A F3E7 C6   50       A          LDB   #GXBSZ   NUMBER OF BYTE X
A F3E9 3D                       MUL
A F3EA 9E   96       A          LDX   GBADX
A F3EC 30   8B       A          LEAX  D,X      TOP OF DATA ADDRESS
A F3EE 9F   BA       A          STX   GBDIX
A F3F0 0F   BC       A          CLR   GBIX     REMAIN, ZERO
A F3F2 39                       RTS
                              *
                              *  POINT COMMAND
                              *
                              *    .COMAR+3  NUMBER OF POINT
                              *    .COMAR+4  POINT DATA
                              *
                              *    STRUCTURE OF POINT DATA
                              *
                              *    0    X1
                              *    2    Y1
                              *    4    COLOR CODE
                              *    5    FUNCTION CODE
                              *
                              *    TOTAL 6 BYTES
                              *
                              *
            F3F3     A POINT  EQU   *
A F3F3 B6   D383     A          LDA   .COMAR+3
A F3F6 27   28    F420          BEQ   POIN80
A F3F8 81   14       A          CMPA  #20
A F3FA 22   25    F421          BHI   POIN90
A F3FC CE   D384     A          LDU   #.COMAR+4
            F3FF     A POIN50 EQU   *
A F3FF 37   30       A          PULU  X,Y
A F401 BD   E446     A          JSR   GAXCHK
A F404 37   06       A          PULU  D
A F406 C1   05       A          CMPB  #5
A F408 24   1A    F424          BHS   POIN92
A F40A DD   56       A          STD   .GCL
A F40C 34   40       A          PSHS  U
A F40E 9F   5A       A          STX   .GX
A F410 109F 58       A          STY   .GY
A F413 BD   E46D     A          JSR   GPDSPI
A F416 BD   E493     A          JSR   GPDSP
A F419 35   40       A          PULS  U
A F41B 7A   D383     A          DEC   .COMAR+3
A F41E 26   DF    F3FF          BNE   POIN50
A F420 39                POIN80 RTS
A F421 7E   E0D7     A POIN90 JMP   ERR65
A F424 7E   E0D4     A POIN92 JMP   ERR64
                              *
                              *  POINT COMMAND END
                              *
                              *
                              *
                              *
                              *  PUT BLOCK1 COMMAND
                              *
                              *    .COMAR+1  PATTERN REMAINED
                              *    .COMAR+3  X1
                              *    .COMAR+5  Y1
                              *    .COMAR+7  X2
                              *    .COMAR+9  Y2
                              *    .COMAR+11 COLOR CODE
                              *    .COMAR+12 FUNCTION CODE
                              *    .COMAR+13 NUMBER OF BYTE OF PATTERN
                              *    .COMAR+14 PATTERN
                              *
            D096     A PBNX   EQU   .WORK
            D098     A PBNY   EQU
            D09A     A PBVRAM EQU
            D09C     A PBSP   EQU
            D09D     A PBEP   EQU
            D09E     A PBLM   EQU
            D09F     A PBRM   EQU
            D0A0     A PBOV   EQU
            D0A1     A PBIB   EQU
            D0A3     A PBDH   EQU
            D0A4     A PBRBEX EQU
            D0A5     A PBRB   EQU
            D0A6     A PBNOT  EQU
            D0A7     A PBBAD  EQU
            D0A9     A PBRD   EQU
            D0AA     A PBSAVE EQU
            D0AB     A PB1BL  EQU
            D0AD     A PB1RE  EQU
            D0AF     A PB1GR  EQU
            D0B1     A PB2ENT EQU
            D0B3     A PBI    EQU
            D0B4     A PBDATA EQU
                              *
            F427     A PBLK1  EQU
A F427 BD   F487     A          JSR
A F42A FC   D38B     A          LDD.
A F42D C1   05       A          CMPB
A F42F 1022 ECA1 E0D4           LBHI
A F433 8E   F46F     A          LDX
A F436 BD   E472     A          JSR
                              *
A F439 96   99       A          LDA
A F43B 4C                       INCA
A F43C 97   B3       A          STA
                              *
            F43E     A PB1050 EQU
A F43E BD   F4EE     A          JSR
A F441 96   9E       A          LDA
A F443 94   B4       A          ANDA
A F445 97   B4       A          STA
A F447 DE   A1       A          LDU
A F449 96   9F       A          LDA
A F44B A4   C9 D0B4  A          ANDA
A F44F A7   C9 D0B4  A          STA
A F453 DE   9A       A          LDU
A F455 30   C4       A          LEAX
A F457 9D   79       A          JSR
A F459 30   C9 4000  A          LEAX
A F45D 9D   7C       A          JSR
A F45F 30   C9 8000  A          LEAX
A F463 9D   7F       A          JSR
A F465 33   C8 50    A          LEAU
A F468 DF   9A       A          STU
A F46A 0A   B3       A          DEC
A F46C 26   D0    F43E          BNE
A F46E 39                       RTS
                              *
A F46F      F60C     A PB1FNC FDB
  A F471    F622     A
A F473      F60C     A          FDB
  A F475    F622     A
A F477      F60C     A          FDB
  A F479    F621     A
A F47B      F621     A          FDB
  A F47D    F622     A
A F47F      F639     A          FDB
  A F481    F621     A
A F483      F60C     A          FDB
  A F485    F622     A
                              *
                              *
                              *  "PBINT"
                              *
                              *      CO-ORD
                              *      AND IN
                              *
                              *
                              *
            F487     A PBINT  EQU
A F487 BD   E406     A          JSR
A F48A 9F   96       A          STX
A F48C DD   98       A          STD
A F48E 9E   5F       A          LDX
A F490 109E 61       A          LDY
A F493 BD   E3E0     A          JSR
A F496 DF   9A       A          STU
A F498 D6   60       A          LDB
A F49A C4   07       A          ANDB
A F49C D7   9C       A          STB
A F49E CE   E396     A          LDU
A F4A1 A6   C5       A          LDA
A F4A3 4A                       DECA
A F4A4 AB   C5       A          ADDA
A F4A6 97   9E       A          STA
A F4A8 D6   64       A          LDB
A F4AA C4   07       A          ANDB
A F4AC D7   9D       A          STB
A F4AE A6   C5       A          LDA
A F4B0 4A                       DECA
A F4B1 43                       COMA
A F4B2 97   9F       A          STA
A F4B4 D6   97       A          LDB
A F4B6 5C                       INCB
A F4B7 C4   07       A          ANDB
A F4B9 D7   A0       A          STB
A F4BB DC   5F       A          LDD
A F4BD BD   E3F6     A          JSR
A F4C0 34   10       A          PSHS
```

```
PBNX+2
PBNY+2
PBVRAM+2
PBSP+1
PBEP+1
PBLM+1
PBRM+1
PBOV+1
PBIB+2
PBDH+1  PBRB EXTEND MSB 0
PBRBEX+1
PBRB+1
PBNOT+1
PBBAD+2
PBRD+1
PBSAVE+1
PB1BL+2
PB1RE+2
PB1GR+2
PB2ENT+2
PBI+1   DATA,B1 BYTE

*
PBINT   INITIALIZE
.COMAR+11 A=COLOR,B=FUNCTION
#5
ERR64
#PB1FNC MAKE FUNCTION ENTRY
FENTRY

PBNY+1

PBI     LOOP COUNTER SET

*
PBSHFT  DATA GET & SHIFT
PBLM
PBDATA
PBDATA  LEFT MASK
PBIB
PBRM
PBDATA,U
PBDATA,U RIGHT MASK
PBVRAM
..VB,U
BLUDSP
..VR,U
REDDSP
..VG,U
GREDSP
GXBSZ,U
PBVRAM
PBI
PB1050

PB1OR,PB1AND PSET

PB1OR,PB1AND PRESET

PB1OR,PB1NOP OR

PB1NOP,PB1AND AND

PB1EOR,PB1NOP EOR

PB1OR,PB1AND NOT



INATE OF WINDOW CHECK AND SORT
ITALIZE SHIFT PROCESS



*
BOXINV
PBNX
PBNY
.BOXX1
.BOXY1
GADDR
PBVRAM  VRAM ADDRESS
.BOXX1+1
#$07
PBSP    BIT NO,START
#BAS
B,U

B,U
PBLM    LEFT MASK
.BOXX2+1
#$07
PBEP    BIT NO,END
B,U

PBRM    RIGHT MASK
PBNX+1

#$07
PBOV    OVER BYTE
.BOXX1
GXOFF
X
```

```
03896A F4C2 DC   63        A       LDD   .BOXX2
03897A F4C4 BD   E3F6      A       JSR   GXOFF
03898A F4C7 1F   10        A       TFR   X,D
03899A F4C9 A3   E1        A       SUBD  ,S++
03900A F4CB DD   A1        A       STD   PBIB    INTERVAL BYTE
03901A F4CD 0F   A3        A       CLR   PBDH    DATA HEAD 0
03902A F4CF 0F   A4        A       CLR   PBRBEX
03903A F4D1 0F   A5        A       CLR   PBRB    REMAINDER BIT 0
03904A F4D3 0F   A6        A       CLR   PBNOT   NOT FLAG CLEAR
03905A F4D5 B6   D38C      A       LDA   .COMAR+12
03906A F4D8 81   05        A       CMPA  #5
03907A F4DA 26   02   F4DE         BNE   PBINT3
03908A F4DC 03   A6        A       COM   PBNOT   NOT FLAG SET
03909A F4DE 23   03   F4E3 PBINT3  BLS   PBINT4
03910A F4E0 7E   E0D4      A       JMP   ERR64   FUNCTION CODE ERROR
03911                 F4E3 A PBINT4 EQU  *
03912A F4E3 B6   D38D      A       LDA   .COMAR+13
03913A F4E6 97   A9        A       STA   PBRD    INPUT DATA,DEC COUNT
03914A F4E8 8E   D38E      A       LDX   #.COMAR+14
03915A F4EB 9F   A7        A       STX   PBBAD   INPUT DATA ADDRESS
03916A F4ED 39                     RTS
03917                           *
03918                           *       'PBSHFT'
03919                           *
03920                           *       PATTERN GET AND SHIFT
03921                           *
03922                           *
03923                 F4EE    A PBSHFT EQU  *
03924A F4EE 96   AA        A       LDA   PBSAVE
03925A F4F0 97   B4        A       STA   PBDATA
03926A F4F2 DC   96        A       LDD   PBNX
03927A F4F4 93   A4        A       SUBD  PBRBEX
03928A F4F6 2B   15   F50D         BMI   PBSH20
03929A F4F8 BD   E3FF      A       JSR   LSRD3
03930A F4FB 8E   D0B4      A       LDX   #PBDATA
03931A F4FE 96   A5        A       LDA   PBRB
03932A F500 27   02   F504         BEQ   PBSH10
03933A F502 30   01        A       LEAX  1,X
03934A F504 8D   48   F54E PBSH10  BSR   PBGET   PATTERN GET
03935A F506 A7   80        A       STA   ,X+
03936A F508 5A                     DECB
03937A F509 2A   F9   F504         BPL   PBSH10
03938A F50B 97   AA        A       STA   PBSAVE  GET END,SAVE
03939
03940                 F50D    A PBSH20 EQU  *
03941A F50D 96   9C        A       LDA   PBSP    SET POS.
03942A F50F 90   A3        A       SUBA  PBDH    N PLACE FOR SHIFT
03943A F511 27   1F   F532         BEQ   PBSH50
03944A F513 2B   0F   F524         BMI   PBSH40
03945                           *
03946A F515 8E   D0B4      A PBSH30 LDX  #PBDATA LEFT SHIFT N PLACE
03947A F518 D6   A2        A       LDB   PBIB+1
03948                 F51A    A PBSH35 EQU  *
03949A F51A 66   80        A       ROR   ,X+
03950A F51C 5A                     DECB
03951A F51D 2A   FB   F51A         BPL   PBSH35
03952A F51F 4A                     DECA
03953A F520 26   F3   F515         BNE   PBSH30
03954A F522 20   0E   F532         BRA   PBSH50
03955A F524 8E   D0B4      A PBSH40 LDX  #PBDATA RIGHT SHIFT N PLACE
03956A F527 D6   A2        A PBSH41 LDB  PBIB+1
03957A F529 5C                     INCB
03958                 F52A    A PBSH45 EQU  *
03959A F52A 69   85        A       ROL   B,X
03960A F52C 5A                     DECB
03961A F52D 2A   FB   F52A         BPL   PBSH45
03962A F52F 4C                     INCA
03963A F530 26   F5   F527         BNE   PBSH41
03964                           *
03965                 F532    A PBSH50 EQU  *
03966A F532 96   A3        A       LDA   PBDH
03967A F534 9B   A0        A       ADDA  PBOV
03968A F536 84   07        A       ANDA  #$07
03969A F538 97   A3        A       STA   PBDH
03970A F53A 40                     NEGA
03971A F53B 84   07        A       ANDA  #$07
03972A F53D 97   A5        A       STA   PBRB
03973A F53F 39                     RTS
03974                           *
03975                           *
03976                           *
03977                           *       'PBGET'
03978                           *
03979                           *       GET PATTERN FROM COMM.AREA
03980                           *       IF PATTERN IS NOTHING, RETURNS "00".
03981                           *
03982                           *
03983                 F540    A PBGET0 EQU  *
03984A F540 BD   E11A      A       JSR   CONTI   NEXT PATTERN
03985A F543 8E   D384      A       LDX   #.COMAR+4
03986A F546 9F   A7        A       STX   PBBAD
03987A F548 B6   D383      A       LDA   .COMAR+3
03988A F54B 97   A9        A       STA   PBRD
03989A F54D                        SKIP2
          A F54D 8C        A       FCB   $8C     CMPX #
03990                 F54E    A PBGET  EQU  *       ENTRY
03991A F54E 34   10        A       PSHS  X
03992A F550 96   A9        A       LDA   PBRD
03993A F552 26   07   F55B         BNE   PBGET2
03994A F554 96   02        A       LDA   ..CONT
03995A F556 2B   E8   F540         BMI   PBGET0
03996A F558 4F                     CLRA
03997A F559 20   08   F563         BRA   PBGET4
03998                 F55B    A PBGET2 EQU  *
03999A F55B 9E   A7        A       LDX   PBBAD
04000A F55D A6   80        A       LDA   ,X+
04001A F55F 9F   A7        A       STX   PBBAD
```

I/O

サブシステム・モニタ ソース・リスト

```
A F561 0A   A9        A        DEC    PBRD
A F563 0D   A6        A PBGET4 TST    PBNOT
A F565 27   01   F568          BEQ    PBGET9
A F567 43                      COMA
A F568               PBGET9 RTSPUL X
A F568 35   90        A        PULS   X,PC
                        *
                        *
                        *
                        *
                        *
                        *   PUT BLOCK2 COMMAND
                        *
                        *    .COMAR+1  PATTERN REMAIND
                        *    .COMAR+3  X1
                        *    .COMAR+5  Y1
                        *    .COMAR+7  X2
                        *    .COMAR+9  Y2
                        *    .COMAR+11 DUMMY
                        *    .COMAR+12 FUNCTION CODE
                        *    .COMAR+13 NUMBER OF BYTE
                        *    .COMAR+14 PATTERN
                        *
                        *
            F56A      A PBLK2  EQU    *
A F56A BD   F487      A        JSR    PBINT
A F56D B6   D38C      A        LDA    .COMAR+12
A F570 81   03        A        CMPA   #3
A F572 26   04   F578          BNE    PB2010
A F574 03   9E        A        COM    PBLM
A F576 03   9F        A        COM    PBRM
            F578      A PB2010 EQU    *
A F578 8E   F64F      A        LDX    #PB2FNC
A F57B 48                      LSLA
A F57C AE   86        A        LDX    A,X
A F57E 9F   B1        A        STX    PB2ENT
A F580 DE   9A        A        LDU    PBVRAM
A F582 8D   0E   F592          BSR    PB2SUB
A F584 DE   9A        A        LDU    PBVRAM
A F586 33   C9 4000   A        LEAU   ..VR,U
A F58A 8D   06   F592          BSR    PB2SUB
A F58C DE   9A        A        LDU    PBVRAM
A F58E 33   C9 8000   A        LEAU   ..VG,U
                        *
            F592      A PB2SUB EQU    *
A F592 0F   A3        A        CLR    PBDH
A F594 0F   A5        A        CLR    PBRB
A F596 96   99        A        LDA    PBNY+1
A F598 4C                      INCA
A F599 97   B3        A        STA    PBI
                        *
            F59B      A PB2050 EQU    *
A F59B BD   F4EE      A        JSR    PBSHFT
A F59E D6   A2        A        LDB    PBIB+1
A F5A0 8E   D0B4      A        LDX    #PBDATA
A F5A3                         VACON
A F5A3 B5   D409      A        BITA   VACC
A F5A6 AD   9F D0B1   A        JSR    [PB2ENT]
A F5AA                         VACOFF
A F5AA B7   D409      A        STA    VACC
A F5AD 33   C8 50     A        LEAU   GXBSZ,U
A F5B0 0A   B3        A        DEC    PBI
A F5B2 26   E7   F59B          BNE    PB2050
A F5B4 39                      RTS
                        *
            F5B5      A PB2LRM EQU    *
A F5B5 96   9E        A        LDA    PBLM
A F5B7 94   B4        A        ANDA   PBDATA
A F5B9 97   B4        A        STA    PBDATA
A F5BB 96   9F        A        LDA    PBRM
A F5BD A4   85        A        ANDA   B,X
A F5BF A7   85        A        STA    B,X
A F5C1 39                      RTS
            F5C2      A PB2SET EQU    *
A F5C2 8D   F1   F5B5          BSR    PB2LRM
A F5C4 96   9E        A        LDA    PBLM
A F5C6 43                      COMA
A F5C7 A4   C4        A        ANDA   ,U
A F5C9 AA   84        A        ORA    ,X
A F5CB A7   84        A        STA    ,X
A F5CD 96   9F        A        LDA    PBRM
A F5CF 43                      COMA
A F5D0 A4   C5        A        ANDA   B,U
A F5D2 AA   85        A        ORA    B,X
A F5D4 A7   85        A        STA    B,X
A F5D6 A6   85        A PB2SE5 LDA    B,X
A F5D8 A7   C5        A        STA    B,U
A F5DA 5A                      DECB
A F5DB 2A   F9   F5D6          BPL    PB2SE5
A F5DD 39                      RTS
                        *
            F5DE      A PB2AND EQU    *
A F5DE 96   9E        A        LDA    PBLM
A F5E0 9A   B4        A        ORA    PBDATA
A F5E2 97   B4        A        STA    PBDATA
A F5E4 96   9F        A        LDA    PBRM
A F5E6 AA   85        A        ORA    B,X
A F5E8 A7   85        A        STA    B,X
A F5EA A6   85        A PB2AN5 LDA    B,X
A F5EC A4   C5        A        ANDA   B,U
A F5EE A7   C5        A        STA    B,U
A F5F0 5A                      DECB
A F5F1 2A   F7   F5EA          BPL    PB2AN5
A F5F3 39                      RTS
                        *
            F5F4      A PB2OR  EQU    *
A F5F4 8D   BF   F5B5          BSR    PB2LRM
A F5F6 A6   85        A PB2OR5 LDA    B,X
A F5F8 AA   C5        A        ORA    B,U
A F5FA A7   C5        A        STA    B,U
A F5FC 5A                      DECB
A F5FD 2A   F7   F5F6          BPL    PB2OR5
A F5FF 39                      RTS
                        *
            F600      A PB2EOR EQU    *
A F600 8D   B3   F5B5          BSR    PB2LRM
A F602 A6   85        A PB2E05 LDA    B,X
A F604 A8   C5        A        EORA   B,U
A F606 A7   C5        A        STA    B,U
A F608 5A                      DECB
A F609 2A   F7   F602          BPL    PB2E05
A F60B 39                      RTS
                        *
                        *
            F60C      A PB1OR  EQU    *
A F60C D6   A2        A        LDB    PBIB+1
A F60E 108E D0B4      A        LDY    #PBDATA
A F612                  LNXOR  VACON
A F612 B5   D409      A        BITA   VACC
A F615 A6   A5        A PB1OR5 LDA    B,Y
A F617 AA   85        A        ORA    B,X
A F619 A7   85        A        STA    B,X
A F61B 5A                      DECB
A F61C 2A   F7   F615          BPL    PB1OR5
A F61E                         VACOFF
A F61E B7   D409      A        STA    VACC
A F621 39               PB1NOP RTS
                        *
            F622      A PB1AND EQU    *
A F622 D6   A2        A        LDB    PBIB+1
A F624 108E D0B4      A        LDY    #PBDATA
A F628                  LNXAND VACON
A F628 B5   D409      A        BITA   VACC
A F62B A6   A5        A PB1AN5 LDA    B,Y
A F62D 43                      COMA
A F62E A4   85        A        ANDA   B,X
A F630 A7   85        A        STA    B,X
A F632 5A                      DECB
A F633 2A   F6   F62B          BPL    PB1AN5
A F635                         VACOFF
A F635 B7   D409      A        STA    VACC
A F638 39               LNXNOP RTS
                        *
            F639      A PB1EOR EQU    *
A F639 D6   A2        A        LDB    PBIB+1
A F63B 108E D0B4      A        LDY    #PBDATA
A F63F                  LNXEOR VACON
A F63F B5   D409      A        BITA   VACC
A F642 A6   A5        A PB1E05 LDA    B,Y
A F644 A8   85        A        EORA   B,X
A F646 A7   85        A        STA    B,X
A F648 5A                      DECB
A F649 2A   F7   F642          BPL    PB1E05
A F64B                         VACOFF
A F64B B7   D409      A        STA    VACC
A F64E 39                      RTS
                        *
A F64F      F5C2      A PB2FNC FDB    PB2SET
A F651      E0D4      A        FDB    ERR64
A F653      F5F4      A        FDB    PB2OR
A F655      F5DE      A        FDB    PB2AND
A F657      F600      A        FDB    PB2EOR
A F659      F5C2      A        FDB    PB2SET
                        *
                        *
                        *   PUT BLOCK1,BLOCK2
                        *
                        *
                        *
                        *
                        *   GCURS ; GRAPHIC CUR
                        *
                        *
                        *
            D096      A GCIX   EQU    .WORK
            D098      A GCIY   EQU    GCIX+2
            D09A      A GCX    EQU    GCIY+2
            D09C      A GCY    EQU    GCX+2
            D09E      A GCCODE EQU    GCY+2
            D09F      A GCCL   EQU    GCCODE+1
            D0A0      A GCSCON EQU    GCCL+1
            D0A1      A GCCONT EQU    GCSCON+1
            D0A3      A GCBCON EQU    GCCONT+2
            D0A4      A GCNCON EQU    GCBCON+1
            D0A5      A GCMCON EQU    GCNCON+1
            D0A6      A GCCURS EQU    GCMCON+1
            D0A9      A GCHAD0 EQU    GCCURS+3
            D0AB      A GCHAD1 EQU    GCHAD0+2
            D0AD      A GCBLUE EQU    GCHAD1+2
            D0AE      A GCRED  EQU    GCBLUE+1
            D0AF      A GCGREE EQU    GCRED+1
            D0B0      A GCBITM EQU    GCGREE+1
            D0B1      A GCWOR  EQU    GCBITM+1
            D0B2      A GCWORK EQU    GCWOR+1
            D0B4      A GCFLG3 EQU    GCWORK+2
            D0B5      A GCFLG5 EQU    GCFLG3+1
            D0B6      A GCCRSI EQU    GCFLG5+1
            D0B9      A GCAREA EQU    GCCRSI+3
            D0E0      A GCSARE EQU    GCAREA+39
                        *
A F65B 7E   E0D7      A GCERR1 JMP    ERR65
                        *
                        *
```

```
END

SOR

 INITIAL CO-ORDINATE Y SAVE AREA
 X CO-ORDINATE SAVE AREA
 Y CO-ORDINATE SAVE AREA
 KEY CODE SAVE AREA
  COLOR CODE SAVE AREA
 SHIFT COUNT SET AREA
  SHIFT DOT NUMBER
  BYTE COUNT SET AREA
  NEEDS CO-ORDINATE NUMBER SET AREA
  NEEDS CO-ORDINATE NUMBER SET AREA
  CURSOR SET AREA
  GCAREA HEAD ADDRESS SET AREA
  CENTER CO-ORDINATE SET AREA HEAD ADDRESS
  BLUE COLOR AREA
  RED COLOR AREA
 GREEN COLOR AREA
  BIT MARK SAVE AREA
  WORK AREA
 WORK AREA
  RETURN FLAG
  INITIAL POSITION SHIFT FLAG
  INITIAL CURSOR SAVE
  SAVE AREA
```

```
04208              F65E      A GCURS  EQU    *
04209A F65E BD     FDDE      A        JSR    KEYR2    KEY RESET 2
04210A F661 B6     D384      A        LDA    .COMAR+4 TEST NEEDS CO- ORDINATE NUMBER
04211A F664 27     F5   F65B          BEQ    GCERR1
04212A F666 81     0A        A        CMPA   #10
04213A F668 22     F1   F65B          BHI    GCERR1
04214                        *
04215A F66A 97     A4        A        STA    GCNCON   STORE NEEDS CO-ORDINATE NUMBER
04216A F66C 97     A5        A        STA    GCMCON
04217                        *
04218A F66E BE     D385      A        LDX    .COMAR+5 TEST INITIAL CO-ORDINATE
04219A F671 10BE   D387      A        LDY    .COMAR+7
04220A F675 BD     E446      A        JSR    GAXCHK
04221                        *
04222                        *
04223              F678      A GC0000 EQU    *
04224A F678 9F     9A        A        STX    GCX      SAVE X CO-ORDINATE
04225A F67A 9F     96        A        STX    GCIX     SAVE INITIAL CO-ORDINATE X
04226A F67C 109F   9C        A        STY    GCY      SAVE Y CO-ORDINATE
04227A F67F 109F   98        A        STY    GCIY     SAVE INITIAL CO-ORDINATE Y
04228A F682 86     FF        A        LDA    #$FF     SET CURSOR
04229A F684 97     A6        A        STA    GCCURS
04230A F686 97     A7        A        STA    GCCURS+1
04231A F688 86     80        A        LDA    #$80
04232A F68A 97     A8        A        STA    GCCURS+2
04233A F68C CC     0001      A        LDD    #1
04234A F68F DD     A1        A        STD    GCCONT   INITIAL DOT NUMBER
04235                        *
04236                        *
04237                        ** SET FLAG
04238                        *
04239A F691 8E     D0E0      A        LDX    #GCSARE
04240A F694 9F     A9        A        STX    GCHADO
04241A F696 8E     D383      A        LDX    #.COMAR+3
04242A F699 9F     AB        A        STX    GCHAD1
04243A F69B 0F     B4        A        CLR    GCFLG3   RETURN FLAG ON
04244                        *
04245                        *
04246                        ** EXTEND COLOR CODE
04247                        *
04248A F69D B6     D383      A        LDA    .COMAR+3 LOAD COLOR CODE
04249A F6A0 84     07        A        ANDA   #$07
04250A F6A2 97     9F        A        STA    GCCL
04251A F6A4 8E     D0AD      A        LDX    #GCBLUE
04252A F6A7 04     9F        A GC0001 LSR    GCCL
04253A F6A9 56                        RORB
04254A F6AA 1D                        SEX
04255A F6AB A7     80        A        STA    ,X+
04256A F6AD 8C     D0B0      A        CMPX   #GCBLUE+3
04257A F6B0 26     F5   F6A7          BNE    GC0001
04258                        *
04259                        *
04260                        *
04261                        *
04262A F6B2 8E     D0B9      A        LDX    #GCAREA  LOAD HEAD ADDRESS OF SAVE AREA
04263A F6B5 BD     F881      A        JSR    GCBDRY   SET BOUNDRY COUNDITION
04264A F6B8 BD     FBED      A        JSR    GCSAVE   VRAM SAVE
04265                        *
04266                        *
04267                        *
04268                        ** SHIFT CURSOR TO INITIAL POSITION
04269                        *
04270A F6BB 9E     9A        A        LDX    GCX      LOAD X CO-ORDINATE
04271A F6BD 1F     10        A        TFR    X,D
04272A F6BF C4     07        A        ANDB   #$07
04273A F6C1 27     05   F6C8          BEQ    GC00
04274A F6C3 D7     A0        A        STB    GCSCON   SET SHIFT COUNT
04275A F6C5 BD     FB1B      A        JSR    GCSFT2
04276              F6C8      A GC00   EQU    *
04277A F6C8 D6     A2        A        LDB    GCCONT+1
04278A F6CA D7     A0        A        STB    GCSCON   SET SHIFT COUNT = 1
04279                        *
04280A F6CC 8E     D0B9      A        LDX    #GCAREA  LOAD HEAD ADDRESS OF GCAREA
04281A F6CF BD     F94E      A        JSR    GCDISP   DISPLY GRAPHIC CURSOR
04282A F6D2 DC     A6        A        LDD    GCCURS   SAVE INITIAL CURSOR
04283A F6D4 DD     B6        A        STD    GCCRSI
04284A F6D6 96     A8        A        LDA    GCCURS+2
04285A F6D8 97     B8        A        STA    GCCRSI+2
04286                        *
04287A F6DA 8E     F72E      A GC0002 LDX    #GCIRQ
04288A F6DD 9F     00        A        STX    ..IRQE   IRQ ENTRY SET
04289A F6DF 1C     EF                 CLI             CLEAR MASK
04290A F6E1 BD     FD8E      A        JSR    KEYGET
04291A F6E4 1A     10                 SEI             MASK
04292A F6E6 8E     0000      A        LDX    #0
04293A F6E9 9F     00        A        STX    ..IRQE   CLEAR ENTRY
04294A F6EB 97     9E        A        STA    GCCODE   SAVE KEY CODE
04295A F6ED 81     0D        A        CMPA   #$0D
04296A F6EF 27     15   F706          BEQ    GC0004   RETURN KEY
04297A F6F1 81     20        A        CMPA   #$20
04298A F6F3 25     4F   F744          BLO    GC000B
04299A F6F5 80     30        A        SUBA   #$30
04300A F6F7 27     05   F6FE          BEQ    GC0003
04301A F6F9 81     09        A        CMPA   #9
04302A F6FB 22     DD   F6DA          BHI    GC0002
04303A F6FD                           SKIP2
     A F6FD        8C        A        FCB    $8C      CMPX #
04304A F6FE 86     0A        A GC0003 LDA    #10
04305A F700 97     A2        A        STA    GCCONT+1
04306A F702 97     A0        A        STA    GCSCON
04307A F704 20     D4   F6DA          BRA    GC0002
04308                        *
04309                        *
04310                        ** RETURN KEY
04311                        *
04312              F706      A GC0004 EQU    *
04313                        *
```

```
                          *
                          ** MEMRIZE CENTER CO-ORDINATE
                          *
A F706 9E    9A     A       LDX    GCX     LOAD X CO-ORDINATE
A F708 109E 9C      A       LDY    GCY     LOAD Y CO-ORDINATE
A F70B DE    AB     A       LDU    GCHAD1
A F70D AF    C1     A       STX    ,U++
A F70F 10AF C1      A       STY    ,U++
A F712 DF    AB     A       STU    GCHAD1  SET CENTER CO-ORDINATE SAVE AREA
                          *
                          *
                          ** VRAM SAVE
                          *
A F714 8E    D0B9   A       LDX    #GCAREA LOAD HEAD ADDRESS OF SAVE DATA
A F717 109E A9      A       LDY    GCHAD0  LOAD HEAD ADDRESS OF SAVE AREA
             F71A   A GC0009 EQU   *
A F71A A6    80     A       LDA    ,X+
A F71C A7    A0     A       STA    ,Y+     SAVE 1-BYTE DATA
A F71E 8C    D0E0   A       CMPX   #GCAREA+39
A F721 26    F7  F71A       BNE    GC0009
A F723 109F A9      A       STY    GCHAD0  SET END ADDDRESS OF SAVE AREA
                          *
A F726 0C    B4     A       INC    GCFLG3  RETURN FLAG OFF
A F728 0A    A4     A       DEC    GCNCON
A F72A 26    AE  F6DA       BNE    GC0002
                          *
                          *
                          ** END GRAPHIC CURSOR COMMAND
                          *
             F72C   A GC000A EQU   *
A F72C 20    06  F734       BRA    GCRTS
                          *
             F72E   A GCIRQ  EQU   *
A F72E 8E    D0B9   A       LDX    #GCAREA
A F731 BD    FA29   A       JSR    GCRSAV  CURRENT CURSOR CLEAR
A F734 9E    A9     A GCRTS  LDX   GCHAD0  STACKED CURSOR CLEAR
A F736 8C    D0E0   A GCIRQ5 CMPX  #GCSARE
A F739 23    08  F743       BLS    GCIRQ9  NO DATA
A F73B 30    88 D9  A       LEAX   -39,X
A F73E BD    FA29   A       JSR    GCRSAV
A F741 20    F3  F736       BRA    GCIRQ5
A F743 39               GCIRQ9 RTS
                          ******   END GCURS
                          *
                          *
                          *
                          *
                          ** CURSOR MOVE KEY
                          *
             F744   A GC000B EQU   *
A F744 0D    B4     A       TST    GCFLG3
A F746 26    0A  F752       BNE    GC000F
A F748 34    02     A       PSHS   A
A F74A 8E    D0B9   A       LDX    #GCAREA
A F74D BD    FA29   A       JSR    GCRSAV
A F750 35    02     A       PULS   A
             F752   A GC000F EQU   *
A F752 0F    B4     A       CLR    GCFLG3
A F754 48                   LSLA
A F755 8E    F769   A       LDX    #GCTABL
A F758 AD    96     A       JSR    [A,X]
                          *
A F75A 8E    D0B9   A       LDX    #GCAREA
A F75D BD    F881   A       JSR    GCBDRY  SET BOUNDARY CONDITION
                          *
A F760 BD    F8ED   A       JSR    GCSAVE  VRAM SAVE
                          *
A F763 BD    F94E   A       JSR    GCDISP  DISPLAY GRAPHIC CURSOR
                          *
A F766 7E    F6DA   A       JMP    GC0002  JUMP KEY GET
                          *
                          *
                          *
             F769   A GCTABL EQU   *
A F769       F7B5   A       FDB    GCNOP   $00
A F76B       F7B5   A       FDB    GCNOP   $01
A F76D       F840   A       FDB    GCL20   $02
A F76F       F7B5   A       FDB    GCNOP   $03
A F771       F7B5   A       FDB    GCNOP   $04
A F773       F7B5   A       FDB    GCNOP   $05
A F775       F7FD   A       FDB    GCR20   $06
A F777       F7B5   A       FDB    GCNOP   $07
A F779       F7B5   A       FDB    GCNOP   $08
A F77B       F7B5   A       FDB    GCNOP   $09
A F77D       F7B5   A       FDB    GCNOP   $0A
A F77F       F7D0   A       FDB    GCHOME  $0B
A F781       F7B5   A       FDB    GCNOP   $0C
A F783       F7B5   A       FDB    GCNOP   $0D
A F785       F7B5   A       FDB    GCNOP   $0E
A F787       F7B5   A       FDB    GCNOP   $0F
A F789       F7B5   A       FDB    GCNOP   $10
A F78B       F7B5   A       FDB    GCNOP   $11
A F78D       F7B5   A       FDB    GCNOP   $12
A F78F       F7B5   A       FDB    GCNOP   $13
A F791       F7B5   A       FDB    GCNOP   $14
A F793       F7B5   A       FDB    GCNOP   $15
A F795       F7B5   A       FDB    GCNOP   $16
A F797       F7B5   A       FDB    GCNOP   $17
A F799       F7B5   A       FDB    GCNOP   $18
A F79B       F7A9   A       FDB    GCUP20  $19
A F79D       F7B6   A       FDB    GCDW20  $1A
A F79F       F7B5   A       FDB    GCNOP   $1B
A F7A1       F806   A       FDB    GCSFTR  $1C
A F7A3       F84B   A       FDB    GCSFTL  $1D
A F7A5       F7E1   A       FDB    GCUP    $1E
A F7A7       F7ED   A       FDB    GCDW    $1F
                          *
```

```
             F7A9   A GCUP20 EQU
A F7A9 DC    9C     A        LDD
A F7AB 83    0014   A        SUBD
A F7AE 2A    03  F7B3        BPL
A F7B0 CC    00C7   A        LDD
A F7B3 DD    9C     A GCUP29 STD
A F7B5 39               GCNOP  RTS
                        *
                        *
                        ** TREAT 20-
                        *
             F7B6   A GCDW20 EQU
A F7B6 CC    00C7   A        LDD
A F7B9 93    9C     A        SUBD
A F7BB 1083  0014   A        CMPD
A F7BF 25    09  F7CA        BCS
A F7C1 DC    9C     A        LDD
A F7C3 C3    0014   A        ADDD
A F7C6 DD    9C     A        STD
A F7C8 20    05  F7CF        BRA
             F7CA   A GC0041 EQU
A F7CA CC    0000   A        LDD
A F7CD DD    9C     A        STD
             F7CF   A GC0042 EQU
A F7CF 39                    RTS
                        *
                        *
                        ** TREAT HOME
                        *
             F7D0   A GCHOME EQU
A F7D0 9E    96     A        LDX
A F7D2 9F    9A     A        STX
A F7D4 9E    98     A        LDX
A F7D6 9F    9C     A        STX
A F7D8 DC    B6     A        LDD
A F7DA DD    A6     A        STD
A F7DC 96    B8     A        LDA
A F7DE 97    A8     A        STA
A F7E0 39                    RTS
                        *
                        *
                        *
                        *
                        ** GCUP : TR
                        *
             F7E1   A GCUP   EQU
A F7E1 DC    9C     A        LDD
A F7E3 93    A1     A        SUBD
A F7E5 2A    03  F7EA        BPL
A F7E7 CC    00C7   A        LDD
             F7EA   A GCUP2  EQU
A F7EA DD    9C     A        STD
A F7EC 39                    RTS
                        *
                        *
                        *
                        ** GCDW : TR
                        *
             F7ED   A GCDW   EQU
A F7ED DC    9C     A        LDD
A F7EF D3    A1     A        ADDD
A F7F1 1083  00C7   A        CMPD
A F7F5 2F    03  F7FA        BLE
A F7F7 CC    0000   A        LDD
             F7FA   A GCDW2  EQU
A F7FA DD    9C     A        STD
A F7FC 39                    RTS
                        *
                        *
                        *
A F7FD CC    0014   A GCR20  LDD
A F800 D7    A0     A        STB
A F802 D3    9A     A        ADDD
A F804 20    04  F80A        BRA
                        *
                        ** GCSFTR :
                        *
             F806   A GCSFTR EQU
A F806 DC    9A     A        LDD
A F808 D3    A1     A        ADDD
A F80A DD    9A     A GCRD0  STD
A F80C 1083  027F   A        CMPD
A F810 2F    09  F81B        BLE
A F812 CC    0000   A        LDD
A F815 DD    9A     A        STD
A F817 C6    01     A        LDB
A F819 20    13  F82E        BRA
             F81B   A GCSFT2 EQU
A F81B D6    A0     A        LDB
A F81D 8E    D0A6   A        LDX
A F820 A6    02     A GCSFT3 LDA
A F822 81    FF     A        CMPA
A F824 27    08  F82E        BEQ
A F826 04    A6     A        LSR
A F828 06    A7     A        ROR
A F82A 06    A8     A        ROR
A F82C 20    0A  F838        BRA
             F82E   A GCSFT4 EQU
A F82E 86    FF     A        LDA
A F830 97    A6     A        STA
A F832 97    A7     A        STA
A F834 86    80     A        LDA
A F836 97    A8     A        STA
             F838   A GCSFT5 EQU
A F838 5A                    DECB
A F839 26    E5  F820        BNE
A F83B D6    A2     A        LDB
```

```
*
GCY
#20
GCUP29
#GYD-1
GCY

DOT DOWN CURSOR KEY

*
#GYD-1
GCY
#20
GC0041
GCY
#20
GCY     ADD 20-DOT TO Y CO-ORDINATE
GC0042
*
#$0000
GCY     SET Y CO-ORDINATE = 0
*

KEY

*
GCIX    LOAD INITIAL CO-ORDINATE X
GCX     SET INITIAL CO-ORDINATE X
GCIY    LOAD INITIAL CO-ORDINATE Y
GCY     SET INITIAL CO-ORDINATE Y
GCCRSI
GCCURS
GCCRSI+2
GCCURS+2

EAT UP CURSOR KEY

*
GCY     LOAD Y CO-ORDINATE
GCCONT
GCUP2
#GYD-1
*
GCY     SET Y CO-ORDINATE = 199

EAT DOWN CURSOR KEY

*
GCY     LOAD Y CO-ORDINATE
GCCONT
#GYD-1
GCDW2
#0
*
GCY     SET Y CO-ORDINATE = 0

#20
GCSCON
GCX
GCRD0
TREAT SHIFT RIGHT CURSOR KEY

*
GCX
GCCONT
GCX
#GXD-1
GCSFT2
#0
GCX     SET X CO-ORDINATE = 0
#1
GCSFT4
*
GCSCON  LOAD SHIFT COUNT
#GCCURS
2,X
#$FF
GCSFT4
GCCURS  SHIFT RIGHT 1 BIT
GCCURS+1
GCCURS+2
GCSFT5
*
#$FF
GCCURS
GCCURS+1
#$80
GCCURS+2
*       DECREMENT SHIFT  COUNT
GCSFT3
GCCONT+1
```

```
F83D D7   A0         A         STB   GCSCON
F83F 39                        RTS
                               *
                               *
F840 C6   14         A GCL20   LDB   #20
F842 D7   A0         A         STB   GCSCON
F844 DC   9A         A         LDD   GCX
F846 83   0014       A         SUBD  #20
F849 20   04    F84F           BRA   GCLD0
                               *
                               *
                               ** GCSFTL : TREAT SHIFT LEFT CURSOR KEY
                               *
                F84B A GCSFTL  EQU   *
F84B DC   9A         A         LDD   GCX      LOAD X CO-ORDINATE
F84D 93   A1         A         SUBD  GCCONT
F84F DD   9A         A GCLD0   STD   GCX
F851 2A   09    F85C           BPL   GCSFT7
F853 CC   027F       A         LDD   #GXD-1
F856 DD   9A         A         STD   GCX      SET X CO-ORDINATE = 639
F858 C6   01         A         LDB   #1
F85A 20   13    F86F           BRA   GCSFT8
                F85C A GCSFT7  EQU   *
F85C D6   A0         A         LDB   GCSCON   LOAD SHIFT COUNT
F85E BE   D0A6       A         LDX   #GCCURS
F861 A6   02         A GCSFT6  LDA   2,X
F863 81   80         A         CMPA  #$80
F865 27   08    F86F           BEQ   GCSFT8
F867 08   A8         A         ASL   GCCURS+2 SHIFT LEFT 1 BIT
F869 09   A7         A         ROL   GCCURS+1
F86B 09   A6         A         ROL   GCCURS
F86D 20   0A    F879           BRA   GCSFT9
                F86F A GCSFT8  EQU   *
F86F 86   01         A         LDA   #$01
F871 97   A6         A         STA   GCCURS
F873 86   FF         A         LDA   #$FF
F875 97   A7         A         STA   GCCURS+1
F877 97   A8         A         STA   GCCURS+2
                F879 A GCSFT9  EQU   *
F879 5A                        DECB
F87A 26   E5    F861           BNE   GCSFT6
F87C D6   A2         A         LDB   GCCONT+1
F87E D7   A0         A         STB   GCSCON
F880 39                        RTS
                               *
                               *
                               *
                               *
                               *
                               ** GCBDRY : SET BOUNDRY CONDITION
                               *
                F881 A GCBDRY  EQU   *
F881 34   10         A         PSHS  X
F883 86   01         A         LDA   #1
F885 97   A3         A         STA   GCBCON   SET INITIAL BYTE NUMBER
F887 9E   9A         A         LDX   GCX      LOAD X CO-ORDINATE
F889 109E 9C         A         LDY   GCY      LOAD Y CO-ORDINATE
F88C BD   E3E0       A         JSR   GADDR
                               *
F88F D7   B0         A         STB   GCBITM   SET BIT MARK
F891 AE   E4         A         LDX   0,S
F893 EF   84         A         STU   0,X
F895 9E   9A         A         LDX   GCX      LOAD X CO-ORDINATE
F897 1F   10         A         TFR   X,D
F899 BD   E3FF       A         JSR   LSRD3
F89C 1F   03         A         TFR   D,U
F89E AE   E4         A         LDX   0,S
F8A0 1183 0000       A         CMPU  #0
F8A4 26   06    F8AC           BNE   GCBDR1
F8A6 86   FF         A         LDA   #$FF     NO DISPLAY
F8A8 A7   02         A         STA   2,X      SET LEFT CONDITION
F8AA 20   04    F8B0           BRA   GCBDR2
                F8AC A GCBDR1  EQU   *
F8AC 6F   02         A         CLR   2,X      SET LEFT CONDITION
F8AE 0C   A3         A         INC   GCBCON
                F8B0 A GCBDR2  EQU   *
F8B0 1183 004F       A         CMPU  #79
F8B4 26   06    F8BC           BNE   GCBDR3
F8B6 86   FF         A         LDA   #$FF     NO DISPLAY
F8B8 A7   03         A         STA   3,X      SET RIGHT CONDITION
F8BA 20   04    F8C0           BRA   GCBDR4
                F8BC A GCBDR3  EQU   *
F8BC 6F   03         A         CLR   3,X      SET RIGHT CONDITION
F8BE 0C   A3         A         INC   GCBCON
                F8C0 A GCBDR4  EQU   *
F8C0 108C 0004       A         CMPY  #4
F8C4 25   06    F8CC           BCS   GCBDR5
F8C6 86   04         A         LDA   #4
F8C8 A7   04         A         STA   4,X      SET UPER BYTE COUNT
F8CA 20   04    F8D0           BRA   GCBDR6
                F8CC A GCBDR5  EQU   *
F8CC 1F   20         A         TFR   Y,D
F8CE E7   04         A         STB   4,X      SET UPER BYTE COUNT
                F8D0 A GCBDR6  EQU   *
F8D0 A6   04         A         LDA   4,X
F8D2 9B   A3         A         ADDA  GCBCON
F8D4 97   A3         A         STA   GCBCON   SET BYTE COUNT
F8D6 CC   00C7       A         LDD   #GYD-1
F8D9 93   9C         A         SUBD  GCY
F8DB E7   05         A         STB   5,X      SET DOWN BYTE COUNT
F8DD C1   04         A         CMPB  #4
F8DF 25   04    F8E5           BCS   GCBDR7
F8E1 C6   04         A         LDB   #4
F8E3 E7   05         A         STB   5,X      SET DOWN BYTE COUNT
                F8E5 A GCBDR7  EQU   *
F8E5 E6   05         A         LDB   5,X
F8E7 DB   A3         A         ADDB  GCBCON
```

サブシステム・モニタ ソース・リスト

```
A F8E9 D7   A3      A           STB     GCBCON  SET TOTAL BYTE COUNNT
A F8EB                          RTSPUL X
  A F8EB 35   90      A           PULS    X,PC
                            *
                            *
                            *
                            *
                            *
                            ** GCSAVE ; SAVE VRAM AREA
                            *
            F8ED    A GCSAVE EQU     *
A F8ED 34   10      A           PSHS    X
A F8EF EE   84      A           LDU     0,X     LOAD VRAM ADDRESS
A F8F1 30   06      A           LEAX    6,X     LOAD HRAD ADDRESS OF SAVE AREA
A F8F3 8D   1C   F911           BSR     GCSAV1
                            *
A F8F5 AE   E4      A           LDX     0,S
A F8F7 EE   84      A           LDU     0,X
A F8F9 30   88 11   A           LEAX    6+11,X
A F8FC 33   C9 4000 A           LEAU    ..VR,U  SET RED ADDRESS
A F900 8D   0F   F911           BSR     GCSAV1
                            *
A F902 AE   E4      A           LDX     0,S
A F904 EE   84      A           LDU     0,X
A F906 30   88 1C   A           LEAX    6+11*2,X
A F909 33   C9 8000 A           LEAU    ..VG,U  SET GREEN ADDRESS
A F90D 8D   02   F911           BSR     GCSAV1
                            *
A F90F                          RTSPUL X
  A F90F 35   90      A           PULS    X,PC
                            *
                            *
                            *
                            *
                            ** GCSAV1 ; VRAM SAVE
                            *
            F911    A GCSAV1 EQU     *
A F911                          VACON
  A F911 B5   D409    A           BITA    VACC
A F914 DF   B2      A           STU     GCWORK  STORE VRAM ADDRESS
A F916 10AE 62      A           LDY     2,S
A F919 A6   24      A           LDA     4,Y     LOAD UPER BYTE COUNT
A F91B C6   50      A           LDB     #80
A F91D 3D                       MUL
A F91E 34   06      A           PSHS    D
A F920 DC   B2      A           LDD     GCWORK
A F922 A3   E1      A           SUBD    ,S++
A F924 1F   03      A           TFR     D,U
                            *
A F926 A6   24      A           LDA     4,Y     LOAD UPER BYTE COUNT
A F928 AB   25      A           ADDA    5,Y     ADD UPER BYTE AND DOWN
            F92A    A GCSA1  EQU     *
A F92A E6   C4      A           LDB     0,U     LOAD VRAM
A F92C E7   80      A           STB     ,X+     SAVE TVRAM TO AREA
A F92E 33   C8 50   A           LEAU    80,U    ADD 80 TO VRAM ADDRESS
A F931 4A                       DECA
A F932 2A   F6   F92A           BPL     GCSA1
                            *
A F934 DE   B2      A           LDU     GCWORK  LOAD VRAM ADDRESS
A F936 A6   22      A           LDA     2,Y     LOAD LEFT CONDITION
A F938 81   FF      A           CMPA    #$FF
A F93A 27   04   F940           BEQ     GCSA2
A F93C E6   5F      A           LDB     -1,U    LOAD VRAM
A F93E E7   80      A           STB     ,X+     SAVE VRAM
            F940    A GCSA2  EQU     *
A F940 A6   23      A           LDA     3,Y     LOAD RIGHT CONDITION
A F942 81   FF      A           CMPA    #$FF
A F944 27   04   F94A           BEQ     GCSA3
A F946 E6   41      A           LDB     1,U     LOAD VRAM
A F948 E7   84      A           STB     0,X     SAVE VRAM
            F94A    A GCSA3  EQU     *
A F94A                          VACOFF
  A F94A B7   D409    A           STA     VACC
A F94D 39                       RTS
                            *
                            *
                            *
                            *
                            *
                            ** GCDISP ; DISPLAY GRAPHIC CURSOR
                            *
            F94E    A GCDISP EQU     *
A F94E 34   10      A           PSHS    X
A F950 10AE E4      A           LDY     0,S
A F953 EE   A4      A           LDU     0,Y     LOAD VRAM ADDRESS
A F955 0D   AD      A           TST     GCBLUE  TST BLUE COLOR
A F957 27   04   F95D           BEQ     GCDIS1
A F959 8D   2C   F987           BSR     GCDOR
A F95B 20   02   F95F           BRA     GCDIS2
            F95D    A GCDIS1 EQU     *
A F95D 8D   74   F9D3           BSR     GCDAND
            F95F    A GCDIS2 EQU     *
A F95F 10AE E4      A           LDY     0,S
A F962 EE   A4      A           LDU     0,Y     LOAD VRAM ADDRESS
A F964 33   C9 4000 A           LEAU    ..VR,U  SET RED ADDRESS
A F968 0D   AE      A           TST     GCRED   TEST RED ADDRESS
A F96A 27   04   F970           BEQ     GCDIS3
A F96C 8D   19   F987           BSR     GCDOR
A F96E 20   02   F972           BRA     GCDIS4
            F970    A GCDIS3 EQU     *
A F970 8D   61   F9D3           BSR     GCDAND
            F972    A GCDIS4 EQU     *
A F972 10AE E4      A           LDY     0,S
A F975 EE   A4      A           LDU     0,Y     LOAD VRAM ADDRESS
A F977 33   C9 8000 A           LEAU    ..VG,U  SET GREEN ADDRESS
A F97B 0D   AF      A           TST     GCGREE  TEST GREEN COLOR
A F97D 27   04   F983           BEQ     GCDIS5
A F97F 8D   06   F987           BSR     GCDOR
A F981 20   02   F985           BRA
            F983    A GCDIS5 EQU
A F983 8D   4E   F9D3           BSR
            F985    A GCDIS6 EQU
A F985                          RTSPUL
  A F985 35   90      A           PULS
                            *
                            *
                            *
                            *
                            ** GCDOR ; TRE
                            *
            F987    A GCDOR  EQU
A F987                          VACON
  A F987 B5   D409    A           BITA
A F98A DF   B2      A           STU
A F98C E6   24      A           LDB
            F98E    A GCD01  EQU
A F98E 96   B0      A           LDA
A F990 5D                       TSTB
A F991 27   0A   F99D           BEQ
A F993 33   C8 B0   A           LEAU
A F996 AA   C4      A           ORA
A F998 A7   C4      A           STA
A F99A 5A                       DECB
A F99B 20   F1   F98E           BRA
                            *
            F99D    A GCD02  EQU
A F99D DE   B2      A           LDU
A F99F E6   25      A           LDB
            F9A1    A GCD03  EQU
A F9A1 96   B0      A           LDA
A F9A3 5D                       TSTB
A F9A4 27   0A   F9B0           BEQ
A F9A6 33   C8 50   A           LEAU
A F9A9 AA   C4      A           ORA
A F9AB A7   C4      A           STA
A F9AD 5A                       DECB
A F9AE 20   F1   F9A1           BRA
            F9B0    A GCD04  EQU
A F9B0 DE   B2      A           LDU
A F9B2 8E   D0A6    A           LDX
A F9B5 6D   22      A           TST
A F9B7 26   06   F9BF           BNE
A F9B9 A6   84      A           LDA
A F9BB AA   5F      A           ORA
A F9BD A7   5F      A           STA
                            *
            F9BF    A GCD05  EQU
A F9BF A6   01      A           LDA
A F9C1 AA   C4      A           ORA
A F9C3 A7   C4      A           STA
                            *
A F9C5 6D   23      A           TST
A F9C7 26   06   F9CF           BNE
A F9C9 A6   02      A           LDA
A F9CB AA   41      A           ORA
A F9CD A7   41      A           STA
            F9CF    A GCD06  EQU
A F9CF                          VACOFF
  A F9CF B7   D409    A           STA
A F9D2 39                       RTS
                            *
                            *
                            *
                            *
                            *
                            ** GCDAND ; TR
                            *
            F9D3    A GCDAND EQU
A F9D3                          VACON
  A F9D3 B5   D409    A           BITA
A F9D6 DF   B2      A           STU
A F9D8 96   B0      A           LDA
A F9DA 43                       COMA
A F9DB 97   B1      A           STA
A F9DD E6   24      A           LDB
            F9DF    A GCDAN1 EQU
A F9DF 5D                       TSTB
A F9E0 27   0C   F9EE           BEQ
A F9E2 96   B1      A           LDA
A F9E4 33   C8 B0   A           LEAU
A F9E7 A4   C4      A           ANDA
A F9E9 A7   C4      A           STA
A F9EB 5A                       DECB
A F9EC 20   F1   F9DF           BRA
            F9EE    A GCDAN2 EQU
A F9EE DE   B2      A           LDU
A F9F0 E6   25      A           LDB
            F9F2    A GCDAN3 EQU
A F9F2 5D                       TSTB
A F9F3 27   0C   FA01           BEQ
A F9F5 96   B1      A           LDA
A F9F7 33   C8 50   A           LEAU
A F9FA A4   C4      A           ANDA
A F9FC A7   C4      A           STA
A F9FE 5A                       DECB
A F9FF 20   F1   F9F2           BRA
            FA01    A GCDAN4 EQU
A FA01 DE   B2      A           LDU
A FA03 8E   D0A6    A           LDX
A FA06 E6   22      A           LDB
A FA08 5D                       TSTB
A FA09 26   07   FA12           BNE
A FA0B A6   84      A           LDA
A FA0D 43                       COMA
A FA0E A4   5F      A           ANDA
A FA10 A7   5F      A           STA
```

```
GCDIS6
*
GCDAND
*
X
X,PC


AT OR

*

VACC
GCWORK  SAVE VRAM ADDRESS IN WORK AREA
4,Y     LOAD UPER BYTE COUNT
*
GCBITM  LOAD BIT MARK

GCD02   TEST UPER BYTE COUNT
-80,U   SUBTRACT 80 FROM VRAM ADDRESS
0,U
0,U     SET VRAM
        DECREMENT UPER BYTE COUNT
GCD01

*
GCWORK  LOAD VRAM ADDRESS
5,Y     LOAD DOWN BYTE COUNT
*
GCBITM

GCD04
80,U    ADD VRAM ADDDRESS AND 80
0,U
0,U     SET VRAM
        DECREMENT DOWN BYTE COUNT
GCD03
*
GCWORK  LOAD VRAM ADDRESS
#GCCURS LOAD HEAD ADDRESS OF CURSOR AREA
2,Y     LOAD LEFT CONDITION
GCD05
0,X
-1,U
-1,U    SET VRAM


*
1,X     SET CENTER CURSOR
0,U
0,U     SET VRAM

3,Y     LOAD RIGHT CONDTION
GCD06
2,X
1,U
1,U     SET VRAM
*

VACC



EAT AND

*

VACC
GCWORK  SAVE VRAM ADDRESS IN WORK AREA
GCBITM  LOAD BIT MARK

GCWOR   STORE COMPLEMENT BIT MARK
4,Y     LOAD UPER BYTE COUNT
*

GCDAN2
GCWOR   LOAD COMPLEMENT BIT MARK
-80,U   SUBTRACT 80 FROM VRAM ADDRESS
0,U
0,U
        DECREMENT UPER BYTE COUNT
GCDAN1
*
GCWORK  LOAD VRAM ADDRESS
5,Y     LOAD DOWN BYTE COUNT
*

GCDAN4
GCWOR   LOAD COMPLEMENT BIT MARK
80,U    ADD 80 AND VRAM ADDRESS
0,U
0,U     SET VRAM

GCDAN3
*
GCWORK  LOAD VRAM ADRESS
#GCCURS LOAD HEAD ADDRESS OF CURSOR AREA
2,Y     LOAD LEFT CONDITION

GCDAN5
0,X

-1,U
-1,U    SET VRAM
```

```
        FA12        A GCDAN5 EQU    *
 FA12 A6    01      A        LDA    1,X
 FA14 43                     COMA
 FA15 A4    C4      A        ANDA   0,U
 FA17 A7    C4      A        STA    0,U     SET VRAM
                     *
 FA19 E6    23      A        LDB    3,Y
 FA1B 5D                     TSTB
 FA1C 26    07  FA25         BNE    GCDAN6
 FA1E A6    02      A        LDA    2,X
 FA20 43                     COMA
 FA21 A4    41      A        ANDA   1,U
 FA23 A7    41      A        STA    1,U     SET VRAM
        FA25        A GCDAN6 EQU    *
 FA25                        VACOFF
 A FA25 B7  D409    A        STA    VACC
 FA28 39                     RTS
                     *
                     *
                     *
                     *
                     *
                     *
                     ** GCRSAV ; RETURN THE CONTENS OF SAVE AREA TO VRAM
                     *
        FA29        A GCRSAV EQU    *
 FA29 34    10      A        PSHS   X
 FA2B EE    84      A        LDU    0,X     LOAD VRAM ADDRESS
 FA2D 30    06      A        LEAX   6,X
 FA2F 8D    1C  FA4D         BSR    GCRSA1
                     *
 FA31 AE    E4      A        LDX    0,S
 FA33 EE    84      A        LDU    0,X     LOAD VRAM ADDRESS
 FA35 30    88 11   A        LEAX   6+11,X
 FA38 33    C9 4000 A        LEAU   ..VR,U  SET RED ADDRESS
 FA3C 8D    0F  FA4D         BSR    GCRSA1
                     *
 FA3E AE    E4      A        LDX    0,S
 FA40 EE    84      A        LDU    0,X
 FA42 30    88 1C   A        LEAX   6+11*2,X
 FA45 33    C9 8000 A        LEAU   ..VG,U  SET GREEN ADDRESS
 FA49 8D    02  FA4D         BSR    GCRSA1
 FA4B                        RTSPUL X
 A FA4B 35  90      A        PULS   X,PC
                     *
                     *
                     *
                     *
                     *
                     ** GCRSA1 ; RETURN THE CONTENS OF SAVE AREA TO VRAM
                     *
        FA4D        A GCRSA1 EQU    *
 FA4D                        VACON
 A FA4D B5  D409    A        BITA   VACC
 FA50 DF    B2      A        STU    GCWORK  STORE VRAM ADDRESS IN WORK AREA
 FA52 10AE 62       A        LDY    2,S
 FA55 A6    24      A        LDA    4,Y     LOAD UPER BYTE COUNT
 FA57 C6    50      A        LDB    #80
 FA59 3D                     MUL
 FA5A 34    06      A        PSHS   D
 FA5C DC    B2      A        LDD    GCWORK
 FA5E A3    E1      A        SUBD   ,S++
 FA60 1F    03      A        TFR    D,U     SET TOP ADDRESS OF CURSOR
                     *
 FA62 A6    24      A        LDA    4,Y     LOAD UPER BYTE COUNT
 FA64 AB    25      A        ADDA   5,Y     ADD UPER BYTE AND DOWN BYTE
 FA66 4C                     INCA
        FA67        A GCRSA2 EQU    *
 FA67 E6    80      A        LDB    ,X+
 FA69 E7    C4      A        STB    0,U
 FA6B 33    C8 50   A        LEAU   80,U    ADD 80 TO VRAM ADDRESS
 FA6E 4A                     DECA
 FA6F 26    F6  FA67         BNE    GCRSA2
                     *
 FA71 DE    B2      A        LDU    GCWORK  LOAD VRAM ADDRESS
 FA73 A6    22      A        LDA    2,Y     LOAD LEFT CONDITION
 FA75 81    FF      A        CMPA   #$FF
 FA77 27    04  FA7D         BEQ    GCRSA3
 FA79 E6    80      A        LDB    ,X+
 FA7B E7    5F      A        STB    -1,U    SET VRAM
        FA7D        A GCRSA3 EQU    *
 FA7D A6    23      A        LDA    3,Y     LOAD RIGHT CONDITION
 FA7F 81    FF      A        CMPA   #$FF
 FA81 27    04  FA87         BEQ    GCRSA4
 FA83 E6    80      A        LDB    ,X+
 FA85 E7    41      A        STB    1,U     SET VRAM
        FA87        A GCRSA4 EQU    *
 FA87                        VACOFF
 A FA87 B7  D409    A        STA    VACC
 FA8A 39                     RTS
                     *
                     *
                     *
                     * PAINT COMMAND
                     *
                     *   .COMAR+3  X CO-ORDINATE
                     *   .COMAR+5  Y CO-ORDINATE
                     *   .COMAR+7  PAINT COLOR CODE
                     *   .COMAR+8  NO. OF COLOR
                     *   .COMAR+9  BOUNDARY COLORS
                     *
                     *
 FA8B                        SORG   LINKLN
        D0A5        A .$ORG  SET    LINKLN
 FA8B                        DS     PASAVE,2 SAVE AREA
        D0A5        A PASAVE EQU    .$ORG
        D0A7        A .$ORG  SET    .$ORG+2
 FA8B                        DS     PABRAN,1 BRANCH CONDITION
```

```
                D0A7     A PABRAN EQU     .$ORG
                D0A8     A .$ORG  SET     .$ORG+1
04945A FABB                       DS      PALIMT,1
                D0A8     A PALIMT EQU     .$ORG
                D0A9     A .$ORG  SET     .$ORG+1
04946A FABB                       DS      PAJOHO,7 CURRENT PAINT DATA
                D0A9     A PAJOHO EQU     .$ORG
                D0B0     A .$ORG  SET     .$ORG+7
04947A FABB                       DS      PADIRY,2 NEXT Y
                D0B0     A PADIRY EQU     .$ORG
                D0B2     A .$ORG  SET     .$ORG+2
04948A FABB                       DS      PAYADD,2
                D0B2     A PAYADD EQU     .$ORG
                D0B4     A .$ORG  SET     .$ORG+2
04949A FABB                       DS      PANXTY,2
                D0B4     A PANXTY EQU     .$ORG
                D0B6     A .$ORG  SET     .$ORG+2
04950A FABB                       DS      PACURY,2
                D0B6     A PACURY EQU     .$ORG
                D0B8     A .$ORG  SET     .$ORG+2
04951A FABB                       DS      PAXOFF,1
                D0B8     A PAXOFF EQU     .$ORG
                D0B9     A .$ORG  SET     .$ORG+1
04952A FABB                       DS      PARL,1
                D0B9     A PARL   EQU     .$ORG
                D0BA     A .$ORG  SET     .$ORG+1
04953A FABB                       DS      PASCND,1
                D0BA     A PASCND EQU     .$ORG
                D0BB     A .$ORG  SET     .$ORG+1
04954A FABB                       DS      PASIMG,1
                D0BB     A PASIMG EQU     .$ORG
                D0BC     A .$ORG  SET     .$ORG+1
04955A FABB                       DS      PATEMP,1
                D0BC     A PATEMP EQU     .$ORG
                D0BD     A .$ORG  SET     .$ORG+1
04956A FABB                       DS      PAVB,1
                D0BD     A PAVB   EQU     .$ORG
                D0BE     A .$ORG  SET     .$ORG+1
04957A FABB                       DS      PAVR,1
                D0BE     A PAVR   EQU     .$ORG
                D0BF     A .$ORG  SET     .$ORG+1
04958A FABB                       DS      PAVG,1
                D0BF     A PAVG   EQU     .$ORG
                D0C0     A .$ORG  SET     .$ORG+1
04959A FABB                       DS      PASTKP,2 STACK PTR.
                D0C0     A PASTKP EQU     .$ORG
                D0C2     A .$ORG  SET     .$ORG+2
04960A FABB                       DS      PASTAK,((.PFTBL-PASTAK)/4*4)
                D0C2     A PASTAK EQU     .$ORG
                D2BE     A .$ORG  SET     .$ORG+(.PFTBL-PASTAK)/4*4
04961A FABB                       DS      PADMMY,0
                D2BE     A PADMMY EQU     .$ORG
                D2BE     A .$ORG  SET     .$ORG+0

04963           D387     A PANBCL EQU     .COMAR+7
04964           D394     A PABCLE EQU     .COMAR+20

04966                    *
04967                    *
04968           FABB     A PAINT  EQU     *
04969A FABB B6  D387     A        LDA     .COMAR+7
04970A FABE 5F                    CLRB
04971A FABF DD  56       A        STD     .GCL
04972A FA91 BD  E456     A        JSR     INITLN
04973A FA94 8E  D2BE     A        LDX     #PADMMY
04974A FA97 9F  C0       A        STX     PASTKP
04975A FA99 CE  D380     A        LDU     #.COMAR
04976A FA9C EC  47       A        LDD     7,U      A=PAINT COLOR,B=NUMBER OF BOUND COLORS.
04977A FA9E C1  08       A        CMPB    #8       IS COLORS GE 8?
04978A FAA0 1022 E63F E0E3        LBHI    ERR69    YES, GO ERROR.
04979A FAA4 5C                    INCB             INCLUDE PAINT COLOR
04980A FAA5 E7  47       A        STB     7,U
04981A FAA7 A7  48       A        STA     8,U
04982A FAA9 8E  D394     A        LDX     #PABCLE EXTEND BOUNDARY COLORS BIT.
04983A FAAC 33  48       A        LEAU    8,U
04984A FAAE 34  04       A        PSHS    B        SET NUMBER OF COLORS.
04985A FAB0 C6  03       A PA0010 LDB     #3
04986A FAB2 A6  C0       A        LDA     ,U+
04987A FAB4 6F  80       A PA0015 CLR     ,X+      ASSUME OFF
04988A FAB6 44                    LSRA
04989A FAB7 24  02   FABB         BCC     PA0017
04990A FAB9 63  1F       A        COM     -1,X     ON
04991A FABB 5A             PA0017 DECB
04992A FABC 26  F6   FAB4         BNE     PA0015
04993A FABE 6A  E4       A        DEC     ,S
04994A FAC0 26  EE   FAB0         BNE     PA0010
04995A FAC2 32  61       A        LEAS    1,S

04997                    * SEARCH THE INITIAL BOUNDARY POSITION.

04999A FAC4 BE  D383     A        LDX     .COMAR+3
05000A FAC7 10BE D385    A        LDY     .COMAR+5
05001A FACB BD  E446     A        JSR     GAXCHK
05002A FACE 9F  AE       A        STX     PAJOHO+5
05003A FAD0 109F AC      A        STY     PAJOHO+3
05004A FAD3 D6  AD       A        LDB     PAJOHO+4 LOAD THE Y COORDINATE.
05005A FAD5 86  50       A        LDA     #80
05006A FAD7 3D                    MUL              LEFT SIDE ADDR OF CURRENT Y.
05007A FAD8 DD  B2       A        STD     PAYADD
05008A FADA BD  FCDC     A        JSR     PASTP1   SEARCH RIGHT SIDE
05009A FADD 9F  AA       A        STX     PAJOHO+1 SET RIGHT END POS.
05010A FADF 9E  AE       A        LDX     PAJOHO+5
05011A FAE1 BD  FCE1     A        JSR     PASTP2   SEARCH LEFT SIDE.
05012A FAE4 9C  AA       A        CMPX    PAJOHO+1 IS THE INIT POS JUST ON BOUNDARY COLOR.
05013A FAE6 26  01   FAE9         BNE     PA0020   NO.
05014A FAE8 39                    RTS
05015A FAE9 9F  AE       A PA0020 STX
05016A FAEB 86  FF       A        LDA
05017A FAED 97  A9       A        STA
05018A FAEF 86  04       A        LDA
05019A FAF1 BD  FC46     A        JSR
05020A FAF4 03  A9       A        COM


05022                    *
05023                    * PAINT
05024                    *
05025           FAF6     A PAPAIN EQU
05026A FAF6 D6  AD       A        LDB
05027A FAF8 86  50       A        LDA
05028A FAFA 3D                    MUL
05029A FAFB DD  B2       A        STD
05030A FAFD 9E  AE       A        LDX
05031A FAFF 30  01       A        LEAX
05032A FB01 DC  AA       A        LDD
05033A FB03 BD  FCE6     A        JSR
05034A FB06 9C  AA       A        CMPX
05035A FB08 102C 00F6 FC02        LBGE
05036A FB0C 33  1F       A        LEAU
05037A FB0E DF  AE       A        STU
05038A FB10 BD  FB82     A        JSR
05039A FB13 9C  AA       A        CMPX
05040A FB15 2C  21   FB38         BGE
05041A FB17 9F  A5       A        STX
05042A FB19 30  01       A        LEAX
05043A FB1B DC  AA       A        LDD
05044A FB1D BD  FCE6     A        JSR
05045A FB20 9C  AA       A        CMPX
05046A FB22 2C  10   FB34         BGE
05047A FB24 DC  AE       A        LDD
05048A FB26 34  06       A        PSHS
05049A FB28 30  1F       A        LEAX
05050A FB2A 9F  AE       A        STX
05051A FB2C 4F                    CLRA
05052A FB2D BD  FC46     A        JSR
05053A FB30 35  10       A        PULS
05054A FB32 9F  AE       A        STX
05055A FB34 9E  A5       A PAPA49 LDX
05056A FB36 9F  AA       A        STX
05057           FB38     A PAPA50 EQU


05059                    *
05060                    * "PASEAR"
05061                    * SEARCH
05062                    *
05063                    *
05064           FB38     A PASEAR EQU
05065A FB38 BD  FC6D     A        JSR
05066                    * NEXT Y
05067A FB3B 9E  AA       A        LDX
05068A FB3D 30  1F       A        LEAX
05069A FB3F 9F  A5       A        STX
05070A FB41 BD  FCDC     A        JSR
05071A FB44 9C  A5       A        CMPX
05072A FB46 26  11   FB59         BNE
05073A FB48 30  1F       A        LEAX
05074A FB4A DC  AE       A        LDD
05075A FB4C BD  FCF7     A        JSR
05076A FB4F 9C  AE       A        CMPX
05077A FB51 102D 00AD FC02        LBLT
05078A FB55 30  01       A        LEAX
05079A FB57 20  41   FB9A         BRA

05081A FB59 9F  A5       A PASE30 STX
05082A FB5B DC  B6       A        LDD
05083A FB5D DD  B2       A        STD
05084A FB5F 9E  AA       A        LDX
05085A FB61 30  01       A        LEAX
05086A FB63 DC  A5       A        LDD
05087A FB65 BD  FCE6     A        JSR
05088A FB68 9C  A5       A        CMPX
05089A FB6A 2C  2C   FB98         BGE
05090A FB6C 9F  A5       A        STX
05091A FB6E BD  FB82     A        JSR
05092A FB71 9F  AA       A        STX
05093A FB73 86  04       A        LDA
05094A FB75 BD  FC46     A        JSR
05095A FB78 9E  A5       A        LDX
05096A FB7A 30  1F       A        LEAX
05097A FB7C 9F  AE       A        STX
05098A FB7E 03  A9       A        COM
05099A FB80 20  B6   FB38         BRA

05101A FB82 9F  67       A PALINL STX
05102A FB84 DC  AC       A        LDD
05103A FB86 DD  69       A        STD
05104A FB88 DD  6D       A        STD
05105A FB8A BD  FCDC     A        JSR
05106A FB8D 34  10       A        PSHS
05107A FB8F 30  1F       A        LEAX
05108A FB91 9F  6B       A        STX
05109A FB93 BD  EECA     A        JSR
05110A FB96                       RTSPUL
       A FB96 35  90     A        PULS

05112A FB98 9E  A5       A PASE40 LDX
05113A FB9A 9F  AA       A PASE45 STX
```

```
YES.
PAJOHO+5
#$FF
PAJOHO
#4
PASTKI  RERUN IS RIGHT SEARCH.
PAJOHO  SET UP-DIRECTION.



*
PAJOHO+4
#80

PAYADD  CRRENT Y
PAJOHO+5 LOAD LEFT X COORDINATE
1,X
PAJOHO+1
PASTP3  SEARCH NOT BOUND TO RIGHT
PAJOHO+1 PASS THE RIGHT END POS.
PASTKO  PASS
-1,X
PAJOHO+5
PALINL  PAINT
PAJOHO+1 FINISH?
PAPA50  YES.
PASAVE  NO.
1,X
PAJOHO+1
PASTP3  SEARCH NOT BOUND TO RIGHT.
PAJOHO+1 OVER THE RIGHT END POS?
PAPA49  YES.
PAJOHO+5
D
-1,X    NO.
PAJOHO+5

PASTKI  RERUN IS PAINT.
X
PAJOHO+5
PASAVE
PAJOHO+1
*



RIGHT END POINT

*
PASEA   INITIALIZE

PAJOHO+1
-1,X    LEFT DOT
PASAVE
PASTP1  SEARCH BOUND TO RIGHT
PASAVE  IS THE LEFT DOT BOUNDARY COLOR?
PASE30  NO.
-1,X    YES.
PAJOHO+5
PASTP4  SEARCH NOT BOUND TO LEFT
PAJOHO+5
PASTKO
1,X     SET RIGHT END POS.
PASE45

PASAVE  SAVE IT
PACURY
PAYADD  CURRENT Y
PAJOHO+1
1,X
PASAVE
PASTP3  SEARCH NOT BOUND TO RIGHT.
PASAVE  CONTINUANS?
PASE40  YES, FIND THE RIGHT END POS.
PASAVE
PALINL  NO, PAINT
PAJOHO+1 SET RIGHT END POS.
#4      RERUN IS RIGHT
PASTKI
PASAVE
-1,X
PAJOHO+5 SET LEFT END POS.
PAJOHO
PASEAR


.LX1
PAJOHO+3
.LY1
.LY2
PASTP1
X
-1,X
.LX2
LINES
X
X,PC

PASAVE
PAJOHO+1
```

```
                              *
                              *
                              *   "PASEAL"
                              *
                              *   SEARCH LEFT END POINT
                              *
             FB9C     A PASEAL EQU   *
FB9C BD      FC6D     A        JSR   PASEA
FB9F 9E      AE       A        LDX   PAJOHO+5
FBA1 30      01       A        LEAX  1,X      RIGHT DOT
FBA3 9F      A5       A        STX   PASAVE
FBA5 BD      FCE1     A        JSR   PASTP2   SEARCH BOUND TO LEFT
FBA8 9C      A5       A        CMPX  PASAVE   IS THE RIGHT DOT BOUNDARY COLOR?
FBAA 26      0F  FBBB          BNE   PAS120   NO.
FBAC 30      01       A        LEAX  1,X
FBAE DC      AA       A        LDD   PAJOHO+1
FBB0 BD      FCE6     A        JSR   PASTP3   YES , SEARCH NOT BOUND TO RIGHT
FBB3 9C      AA       A        CMPX  PAJOHO+1
FBB5 2E      4B  FC02          BGT   PASTKO
FBB7 30      1F       A        LEAX  -1,X     SET RIGHT END POS
FBB9 20      3B  FBF6          BRA   PAS145

FBBB 9F      A5       A PAS120 STX   PASAVE
FBBD DC      B6       A        LDD   PACURY   CURRENT Y
FBBF DD      B2       A        STD   PAYADD
FBC1 9E      AE       A        LDX   PAJOHO+5
FBC3 30      1F       A        LEAX  -1,X
FBC5 DC      A5       A        LDD   PASAVE
FBC7 BD      FCF7     A        JSR   PASTP4   SEARCH NOT BOUND TO LEFT
FBCA 9C      A5       A        CMPX  PASAVE   CONTINUANS?
FBCC 2F      26  FBF4          BLE   PAS140   YES, FIND THE LEFT END POS.
FBCE 9F      A5       A        STX   PASAVE
FBD0 9F      6B       A        STX   .LX2
FBD2 DC      AC       A        LDD   PAJOHO+3
FBD4 DD      69       A        STD   .LY1
FBD6 DD      6D       A        STD   .LY2
FBD8 BD      FCE1     A        JSR   PASTP2
FBDB 9F      AE       A        STX   PAJOHO+5
FBDD 30      01       A        LEAX  1,X
FBDF 9F      67       A        STX   .LX1
FBE1 BD      EECA     A        JSR   LINES    PAINT
FBE4 86      08       A        LDA   #8       RERUN IS LEFT
FBE6 BD      FC46     A        JSR   PASTKI   SAVE TO STACK
FBE9 9E      A5       A        LDX   PASAVE
FBEB 30      01       A        LEAX  1,X
FBED 9F      AA       A        STX   PAJOHO+1 SET RIGHT END POINT
FBEF 03      A9       A        COM   PAJOHO
FBF1 7E      FB38     A        JMP   PASEAR


FBF4 9E      A5       A PAS140 LDX   PASAVE
FBF6 9F      AE       A PAS145 STX   PAJOHO+5 SET LEFT END POS.
FBF8 D6      B1       A        LDB   PADIRY+1
FBFA D7      AD       A        STB   PAJOHO+4


                              *
                              *
                              * END JUDGMENT
                              *
FBFC 9C      AA       A        CMPX  PAJOHO+1
FBFE 102D    FEF4 FAF6         LBLT  PAPAIN
                              *
                              * TAKE OUT INFORMATION FROM STACK
                              *
             FC02     A PASTKO EQU   *
FC02 DE      C0       A        LDU   PASTKP   LOAD STACK POINTER
FC04 1183    D2BE     A        CMPU  #PADMMY
FC08 26      01  FC0B          BNE   PAS010
FC0A 39                        RTS            END "PAINT" COMMAND
FC0B 0F      A9       A PAS010 CLR   PAJOHO
FC0D A6      C0       A        LDA   ,U+
FC0F 2A      02  FC13          BPL   PAS020
FC11 03      A9       A        COM   PAJOHO
FC13 98      A9       A PAS020 EORA  PAJOHO
FC15 34      02       A        PSHS  A
FC17 84      CF       A        ANDA  #$CF
FC19 5F                        CLRB
FC1A 44                        LSRA
FC1B 59                        ROLB
FC1C 44                        LSRA
FC1D 59                        ROLB
FC1E C1      03       A        CMPB  #$03     LEFT X IS -1?
FC20 25      02  FC24          BLO   PAS030   NO.
FC22 C6      FF       A        LDB   #$FF     YES.
FC24 D7      AE       A PAS030 STB   PAJOHO+5 X2 HIGH
FC26 E6      C0       A        LDB   ,U+      X1 LOW
FC28 DD      AA       A        STD   PAJOHO+1 X1
FC2A A6      C0       A        LDA   ,U+
FC2C 97      AF       A        STA   PAJOHO+6 X2 LOW
FC2E A6      C0       A        LDA   ,U+
FC30 97      AD       A        STA   PAJOHO+4 Y
FC32 DF      C0       A        STU   PASTKP   SAVE STACK POINTER
FC34 35      02       A        PULS  A
FC36 84      30       A        ANDA  #$30
FC38 44                        LSRA
FC39 44                        LSRA
FC3A 44                        LSRA
FC3B 8E      FC40     A        LDX   #PATABL
FC3E 6E      96       A        JMP   [A,X]
                              *
             FC40     A PATABL EQU   *
FC40         FAF6     A        FDB   PAPAIN   CHECK BOUNDRY COLOR
FC42         FB38     A        FDB   PASEAR   SEARH RIGHT END POINT
FC44         FB9C     A        FDB   PASEAL   SEARCH LEFT END POINT
                              *
                              *
```

```
                           *     "PASTKI"
                           *
                           *     SAVE INFORMATION IN STACK
                           *
               FC46      A PASTKI EQU   *
A FC46 97      A7        A        STA   PABRAN
A FC48 DE      C0        A        LDU   PASTKP  LOAD STACK POINTER
A FC4A 1183    D0C2      A        CMPU  #PASTAK
A FC4E 27      1C   FC6C          BEQ   PASTKR
                           *
A FC50 96      AD        A PAST10 LDA   PAJOHO+4 Y
A FC52 A7      C2        A        STA   ,-U
A FC54 96      AF        A        LDA   PAJOHO+6
A FC56 A7      C2        A        STA   ,-U
A FC58 96      AB        A        LDA   PAJOHO+2
A FC5A A7      C2        A        STA   ,-U
A FC5C 96      AA        A        LDA   PAJOHO+1
A FC5E 9A      A7        A        ORA   PABRAN
A FC60 D6      AE        A        LDB   PAJOHO+5
A FC62 54                         LSRB
A FC63 49                         ROLA
A FC64 54                         LSRB
A FC65 49                         ROLA
A FC66 98      A9        A        EORA  PAJOHO
A FC68 A7      C2        A        STA   ,-U
A FC6A DF      C0        A        STU   PASTKP
A FC6C 39                  PASTKR RTS
                           *
                           *
                           *     "PASEA"
                           *
                           *     SEARCH CONTINUOUS END POINT
                           *
               FC6D      A PASEA  EQU   *
A FC6D DE      AC        A        LDU   PAJOHO+3
A FC6F 33      41        A        LEAU  1,U
A FC71 D6      AD        A        LDB   PAJOHO+4
A FC73 96      A9        A        LDA   PAJOHO
A FC75 26      04   FC7B          BNE   PASE10
A FC77 5A                         DECB
A FC78 33      5E        A        LEAU  -2,U
A FC7A                            SKIP1
  A FC7A       21        A        FCB   $21
A FC7B 5C                  PASE10 INCB
A FC7C C1      C8        A        CMPB  #GYD
A FC7E 25      05   FC85          BLO   PASE20
A FC80 32      62        A        LEAS  2,S
A FC82 7E      FC02      A        JMP   PASTK0
A FC85 86      50        A PASE20 LDA   #80
A FC87 3D                         MUL
A FC88 DD      B4        A        STD   PANXTY
A FC8A DD      B2        A        STD   PAYADD
A FC8C DF      B0        A        STU   PADIRY
A FC8E 86      50        A        LDA   #80
A FC90 D6      AD        A        LDB   PAJOHO+4
A FC92 3D                         MUL
A FC93 DD      B6        A        STD   PACURY
A FC95 39                         RTS


                           *
                           * 'PABGET'
                           * INPUT AND REFERENCE
                           * PAYADD,PAXOFF,PANBCL
                           * PABCLE,PASCND
                           *
                           * OUTPUT
                           * PASIMG SEARCHING COLORS IMG.
                           *           ACCB IS SAME IT.
                           *
                           * WORK
                           * PATEMP,PAVB,PAVR,PAVG
                           *
               FC96      A PABGET EQU   *
A FC96 DE      B2        A        LDU   PAYADD
A FC98 96      B8        A        LDA   PAXOFF
A FC9A 33      C6        A        LEAU  A,U
A FC9C                            VACON
  A FC9C B5    D409      A        BITA  VACC
A FC9F 12                         NOP
A FCA0 E6      C9 8000   A        LDB   ..VG,U
A FCA4 D7      BF        A        STB   PAVG
A FCA6 E6      C9 4000   A        LDB   ..VR,U
A FCAA A6      C4        A        LDA   ..VB,U
A FCAC                            VACOFF
  A FCAC B7    D409      A        STA   VACC
A FCAF DD      BD        A        STD   PAVB
A FCB1 0F      BB        A        CLR   PASIMG
A FCB3 B6      D387      A        LDA   PANBCL
A FCB6 8E      D394      A        LDX   #PABCLE
A FCB9 D6      BD        A PABG50 LDB   PAVB
A FCBB E8      84        A        EORB  ,X
A FCBD D7      BC        A        STB   PATEMP
A FCBF D6      BE        A        LDB   PAVR
A FCC1 E8      01        A        EORB  1,X
A FCC3 DA      BC        A        ORB   PATEMP
A FCC5 D7      BC        A        STB   PATEMP
A FCC7 D6      BF        A        LDB   PAVG
A FCC9 E8      02        A        EORB  2,X
A FCCB DA      BC        A        ORB   PATEMP
A FCCD 53                         COMB
A FCCE DA      BB        A        ORB   PASIMG
A FCD0 D7      BB        A        STB   PASIMG
A FCD2 30      03        A        LEAX  3,X
A FCD4 4A                         DECA
A FCD5 26      E2   FCB9          BNE   PABG50
```

```
A FCD7 DB      BA        A
A FCD9 D7      BB        A
A FCDB 39


A FCDC CE      0100      A PASTP1
A FCDF 20      12   FCF3
A FCE1 CE      FF00      A PASTP2
A FCE4 20      1E   FD04
A FCE6 CE      01FF      A PASTP3
A FCE9 C3      0008      A
A FCEC BD      E3FF      A
A FCEF C1      50        A
A FCF1 2F      13   FD06
A FCF3 C6      50        A PASTRM
A FCF5 20      0F   FD06
A FCF7 CE      FFFF      A PASTP4
A FCFA 83      0008      A
A FCFD BD      E3FF      A
A FD00 C1      FF        A
A FD02 2C      02   FD06
A FD04 C6      FF        A PASTLM
A FD06 DF      B9        A PA.SEA
A FD08 D7      A8        A
A FD0A 8C      FFFF      A
A FD0D 2F      3F   FD4E
A FD0F 8C      0280      A
A FD12 2C      31   FD45
A FD14 1F      10        A
A FD16 C4      07        A
A FD18 34      04        A
A FD1A 1F      10        A
A FD1C BD      E3FF      A
A FD1F D7      B8        A
A FD21 BD      FC96      A
A FD24 35      04        A
A FD26 8E      E396      A
A FD29 E6      85        A
A FD2B 96      B9        A
A FD2D 2B      03   FD32
A FD2F 58
A FD30 5A
A FD31
  A FD31       8C        A
A FD32 5A                  PA.S20
A FD33 53

A FD34 D4      BB        A
A FD36 D7      BB        A
A FD38 26      1F   FD59
A FD3A 96      B8        A PA.S25
A FD3C D6      B9        A
A FD3E 2B      09   FD49
A FD40 4C
A FD41 91      A8        A
A FD43 2D      0D   FD52
A FD45 8E      0280      A PA.RMX
A FD48 39
A FD49 4A                  PA.S30
A FD4A 91      A8        A
A FD4C 2E      04   FD52
A FD4E 8E      FFFF      A PA.LMX
A FD51 39
A FD52 97      B8        A PA.S40
A FD54 BD      FC96      A
A FD57 27      E1   FD3A
A FD59 86      08        A PA.S80
A FD5B D6      B8        A
A FD5D 3D
A FD5E 1F      01        A
A FD60 D6      BB        A
A FD62 96      B9        A
A FD64 2B      07   FD6D
A FD66 58                  PA.S83
A FD67 25      0B   FD74
A FD69 30      01        A
A FD6B 20      F9   FD66
A FD6D 30      1F        A PA.S85
A FD6F 54
A FD70 24      FB   FD6D
A FD72 30      08        A
A FD74 39                  PA.S90
A FD75 39                  PA.RTS
                           *
                           *  INKEY
                           *
               FD76      A INKEY
A FD76 B6      D383      A
A FD79 85      02        A
A FD7B 26      08   FD85
A FD7D 85      04        A
A FD7F 27      06   FD87
A FD81 D6      03        A
A FD83 2B      02   FD87
A FD85 0F      03        A INKE1
A FD87 44                  INKE3
A FD88 8D      06   FD90
A FD8A FD      D383      A
A FD8D 39
                           *
A FD8E 1A      01          KEYGET
               FD90      A KEYIN
                           *
                           *  INPUT
                           *
                           *
```

```
EORB  PASCND
STB   PASIMG
RTS



LDU   #$0100   TO RIGHT,BOUND
BRA   PASTRM
LDU   #$FF00   TO LEFT,BOUND
BRA   PASTLM
LDU   #$01FF   TO RIGHT,NOT BOUND
ADDD  #8
JSR   LSRD3
CMPB  #80
BLE   PA.SEA
LDB   #80
BRA   PA.SEA
LDU   #$FFFF   TO LEFT, NOT BOUND
SUBD  #8
JSR   LSRD3
CMPB  #-1
BGE   PA.SEA
LDB   #-1
STU   PARL     SET SEARCH CONDITION
STB   PALIMT
CMPX  #-1
BLE   PA.LMX
CMPX  #GXD
BGE   PA.RMX
TFR   X,D
ANDB  #$07
PSHS  B
TFR   X,D
JSR   LSRD3
STB   PAXOFF
JSR   PABGET
PULS  B
LDX   #BAS
LDB   B,X
LDA   PARL
BMI   PA.S20
LSLB
DECB
SKIP2
FCB   $8C      CMPX #
DECB
COMB

ANDB  PASIMG
STB   PASIMG
BNE   PA.S80
LDA   PAXOFF
LDB   PARL
BMI   PA.S30
INCA
CMPA  PALIMT
BLT   PA.S40
LDX   #GXD
RTS
DECA
CMPA  PALIMT
BGT   PA.S40
LDX   #-1
RTS
STA   PAXOFF
JSR   PABGET
BEQ   PA.S25
LDA   #8
LDB   PAXOFF
MUL
TFR   D,X
LDB   PASIMG
LDA   PARL
BMI   PA.S85
LSLB
BCS   PA.S90
LEAX  1,X
BRA   PA.S83
LEAX  -1,X
LSRB
BCC   PA.S85
LEAX  8,X
RTS
RTS

  COMMAND

EQU   *
LDA   CMDPRM
BITA  #$02     RESET1?
BNE   INKE1    YES.
BITA  #$04     NO. RESET2?
BEQ   INKE3    NO. NORESET
LDB   KEYDAT   YES. PF?
BMI   INKE3    YES. NO RESET
CLR   KEYDAT   RESET
LSRA           WAIT SET TO C
BSR   KEYIN
STD   CMDPRM   KEYDATA
RTS

SEC            WAIT SET
EQU   *

  C FLAG : WAIT FOR DATA
          IF C IS SET.
```

```
05427                                   * OUTPUT
05428                                   *          ACCA : CODE
05429                                   *          ACCB : 0 IS NO-DATA
05430                                   *
05431A FD90                     KEYINO SEF    FAST   IRQ MASK
       A FD90 1A    40       A         ORCC   #$40
05432A FD92 D6      03       A         LDB    KEYDAT
05433A FD94 26      06   FD9C          BNE    KEYIN1  SOME DATA EXIST.
05434A FD96 24      1E   FDB6          BCC    KEYIN9  RETURN IF NO WAIT IS REQUESTED.
05435A FD98                            CLF    REMOVE  MASK
       A FD98 1C    BF       A         ANDCC  #$BF
05436A FD9A 20      F4   FD90          BRA    KEYINO
05437A FD9C 2A      12   FDB0 KEYIN1   BPL    KEYIN5  PF? NO.
05438A FD9E 0A      04       A         DEC    KEYDAT+1 YES .REMAIN ?
05439A FDA0 2A      06   FDA8          BPL    KEYIN2  YES.
05440A FDA2 86      00       A         LDA    #$00
05441A FDA4 97      03       A         STA    KEYDAT
05442A FDA6 20      E8   FD90          BRA    KEYINO
05443A FDA8 DE      05       A KEYIN2  LDU    PFIX
05444A FDAA A6      C0       A         LDA    ,U+
05445A FDAC DF      05       A         STU    PFIX
05446A FDAE 20      04   FDB4          BRA    KEYIN7
05447A FDB0 96      04       A KEYIN5  LDA    KEYDAT+1
05448A FDB2 0F      03       A         CLR    KEYDAT
05449A FDB4 C6      01       A KEYIN7  LDB    #1
05450A FDB6                     KEYIN9 CLF    MASK    OFF
       A FDB6 1C    BF       A         ANDCC  #$BF
05451A FDB8 39                         RTS
05452                                   *
05453                                   * KEY DATA GET
05454                                   *
05455               FDB9       A KEYIRQ EQU   *       FAST IRQ ENTRY
05456A FDB9 34      16       A         PSHS   D,X
05457A FDBB FC      D400     A         LDD    KCOD1
05458A FDBE 8A      01       A         ORA    #$01
05459A FDC0 DD      03       A         STD    KEYDAT
05460A FDC2 2A      0F   FDD3          BPL    KEYIR5  DATA CODE
05461A FDC4 86      10       A         LDA    #PFSZ   PFCODE
05462A FDC6 3D                         MUL
05463A FDC7 8E      D2B0     A         LDX    #.PFTBL-PFSZ
05464A FDCA 3A                         ABX
05465A FDCB A6      80       A         LDA    ,X+
05466A FDCD 2B      07   FDD6          BMI    KEYIR6  INTERRUPT
05467A FDCF 97      04       A         STA    KEYDAT+1 STRING LENGTH
05468A FDD1 9F      05       A         STX    PFIX    PF STR PTR
05469A FDD3 35      16       A KEYIR5  PULS   D,X
05470A FDD5 3B                         RTI
05471A FDD6 96      04       A KEYIR6  LDA    KEYDAT+1
05472A FDD8 97      07       A         STA    .PFIRQ
05473A FDDA 0F      03       A         CLR    KEYDAT  EMPTY SET
05474A FDDC 20      F5   FDD3          BRA    KEYIR5
05475                                   *
05476               FDDE       A KEYR2  EQU   *       RESET2
05477A FDDE 0D      03       A         TST    KEYDAT
05478A FDE0 2B      02   FDE4          BMI    KEYR2R
05479A FDE2 0F      03       A         CLR    KEYDAT
05480A FDE4 39                  KEYR2R RTS
05481                                   *
05483                                   * FUNCTION KEY COMMAND
05484                                   *
05485               FDE5       A FUNKEY EQU   *
05486A FDE5                            SEF    MASK
       A FDE5 1A    40       A         ORCC   #$40
05487A FDE7 F6      D383     A         LDB    .COMAR+3 IS THE KEY NUMBER .GT. ZERO?
05488A FDEA 27      28   FE14          BEQ    FUNKE9  NO, GIVE ERROR.
05489A FDEC C1      0A       A         CMPB   #10
05490A FDEE 22      24   FE14          BHI    FUNKE9  YES, IS THE KEY NUMBER .LE. 10?
05491A FDF0 B6      D384     A         LDA    COMAR+4
05492A FDF3 81      0F       A         CMPA   #15
05493A FDF5 22      1D   FE14          BHI    FUNKE9
05494A FDF7 8D      27   FE20          BSR    FUNADR
05495A FDF9 1F      89       A         TFR    A,B
05496A FDFB 58                         LSLB
05497A FDFC 68      84       A         LSL    ,X
05498A FDFE 56                         RORB
05499A FDFF E7      80       A         STB    ,X+
05500A FE01 C6      0F       A         LDB    #PFSZ-1
05501A FE03 8D      12   FE17          BSR    CLRARA
05502A FE05 4D                         TSTA
05503A FE06 27      52   FE5A          BEQ    PFINDC
05504A FE08 CE      D385     A         LDU    #.COMAR+5
05505               FE0B       A FUNKE5 EQU   *
05506A FE0B E6      C0       A         LDB    ,U+
05507A FE0D E7      80       A         STB    ,X+
05508A FE0F 4A                         DECA
05509A FE10 26      F9   FE0B          BNE    FUNKE5
05510A FE12 20      46   FE5A          BRA    PFINDC
05511A FE14 7E      E0E0     A FUNKE9  JMP    ERR68
05512                                   *
05513               FE17       A CLRARA EQU   *
05514A FE17 34      10       A         PSHS   X
05515A FE19 6F      80       A CLRAR5  CLR    ,X+
05516A FE1B 5A                         DECB
05517A FE1C 26      FB   FE19          BNE    CLRAR5
05518A FE1E                            RTSPUL X
       A FE1E 35    90       A         PULS   X,PC
05519                                   *
05520A FE20 58                  FUNADR LSLB
05521A FE21 58                         LSLB
05522A FE22 58                         LSLB
05523A FE23 58                         LSLB
05524A FE24 8E      D2B0     A         LDX    #.PFTBL-PFSZ
05525A FE27 3A                         ABX
05526A FE28 39                         RTS
05527                                   *
05529                                   * GET STRING OF FUNCTION-KEY COMMAND
05530                                   *
```

```
05531             FE29     A GSTRPF EQU    *
05532A FE29 CE    D383     A        LDU    #.COMAR+3 * LOAD COMMON ADDRESS
05533A FE2C E6    C0       A        LDB    ,U+
05534A FE2E 27    E4   FE14         BEQ    GSTRP9
05535A FE30 C1    0A       A        CMPB   #10
05536A FE32 22    E0   FE14         BHI    GSTRP9
05537A FE34 8D    EA   FE20         BSR    FUNADR
05538A FE36 E6    80       A        LDB    ,X+
05539A FE38 C4    7F       A        ANDB   #$7F
05540A FE3A 86    10       A        LDA    #PFSZ
05541A FE3C E7    C0       A GSTRP5 STB    ,U+
05542A FE3E E6    80       A        LDB    ,X+
05543A FE40 4A                      DECA        * DECREMENT
05544A FE41 26    F9   FE3C         BNE    GSTRP5
05545A FE43 39                      RTS
05546             FE14     A GSTRP9 EQU    FUNKE9
05547                             *
05548                             *  INTERRUPT CONTROL FOR PROGRAMABLE FUNCTION KEY
05549                             *
05550             FE44     A PFINT  EQU    *
05551A FE44                         SEF
     A FE44 1A    40       A        ORCC   #$40
05552A FE46 FC    D383     A        LDD    CMDPRM   FLAGS
05553A FE49 8E    D2C0     A        LDX    #.PFTBL PF TABLE PTR TO X
05554A FE4C 68    84       A PFINT5 LSL    ,X       ENABLE BIT TO C-FLAG
05555A FE4E 44                      LSRA
05556A FE4F 56                      RORB
05557A FE50 66    84       A        ROR    ,X       SET TO MSB OF COUNTER BYTE
05558A FE52 30    88 10    A        LEAX   PFSZ,X  NEXT PF
05559A FE55 8C    D360     A        CMPX   #.PFTBL+PFSZ*10
05560A FE58 25    F2   FE4C         BLO    PFINT5
05561A FE5A                  PFINDC CLF    *        ENABLE THE KEY INTERRUPT.
     A FE5A 1C    BF       A        ANDCC  #$BF
05562A FE5C 7E    EDBE     A        JMP    INDICT  DISPLAY PF STRINGS.
05563                               TTL    TIMER - TIMER CODE -
05565                             *
05566                             *  SET TIMER COMMAND
05567                             *
05568             FE5F     A SETTIM EQU    *
05569A FE5F 03    08       A        COM    .TMAC    NMI MASK
05570A FE61 8E    D383     A        LDX    #CMDPRM
05571A FE64 108E  D009     A        LDY    #.$TC
05572A FE68 EC    81       A        LDD    ,X++
05573A FE6A 44                      LSRA
05574A FE6B 24    02   FE6F         BCC    SETTM1
05575A FE6D E7    A4       A        STB    ,Y
05576A FE6F 31    21       A SETTM1 LEAY   1,Y
05577A FE71 C6    04       A        LDB    #4
05578A FE73 44               SETTM3 LSRA
05579A FE74 24    08   FE7E         BCC    SETTM5
05580A FE76 EE    84       A        LDU    ,X
05581A FE78 EF    A4       A        STU    ,Y
05582A FE7A EE    02       A        LDU    2,X
05583A FE7C EF    22       A        STU    2,Y
05584A FE7E 30    04       A SETTM5 LEAX   4,X
05585A FE80 31    24       A        LEAY   4,Y
05586A FE82 5A                      DECB
05587A FE83 26    EE   FE73         BNE    SETTM3
05588A FE85 20    0E   FE95         BRA    TMCT
05589                             *
05590                             *  READ TIMER COMMAND
05591                             *
05592             FE87     A REDTIM EQU    *
05593A FE87 03    08       A        COM    .TMAC
05594A FE89 8E    D009     A        LDX    #.$TC
05595A FE8C 108E  D384     A        LDY    #CMDPRM+1
05596A FE90 C6    11       A        LDB    #1+4*4
05597A FE92 BD    E44E     A        JSR    MOVE
05598                             *
05599             FE95     A TMCT   EQU    *        COMMAND TERMINATION
05600A FE95 03    08       A        COM    .TMAC    SET & TEST
05601A FE97 27    08   FEA1         BEQ    TMCT99   NO NMI
05602A FE99                         SEF
     A FE99 1A    40       A        ORCC   #$40
05603A FE9B 8D    0E   FEAB         BSR    TIMER    NMI
05604A FE9D                         CLF
     A FE9D 1C    BF       A        ANDCC  #$BF
05605A FE9F 0F    08       A        CLR    .TMAC
05606A FEA1 39               TMCT99 RTS
05607                             *
05608                             *  20MS TIMER
05609                             *
05610             FEA2     A TIMNMI EQU    *
05611A FEA2 96    08       A        LDA    .TMAC
05612A FEA4 26    02   FEA8         BNE    TIMRTI
05613A FEA6 8D    03   FEAB         BSR    TIMER
05614A FEA8 0F    08       A TIMRTI CLR    .TMAC
05615A FEAA 3B                      RTI
05616                             *
05617                             *  TIMER PACKAGE
05618                             *
05619             FEAB     A TIMER  EQU    *
05620A FEAB 96    1A       A        LDA    .CURSR   IS HERE CURSOR DISPLAY STATE?
05621A FEAD 27    1B   FECA         BEQ    CURTM9   NO.
05622A FEAF 2B    19   FECA         BMI    CURTM9
05623A FEB1 0A    1B       A        DEC    .CURTM   DECREMENT CURSOR INTERVAL.
05624A FEB3 26    15   FECA         BNE    CURTM9   NO CHANGE STATE
05625A FEB5 A6    9F D050 A         LDA    [.CBAD]
05626A FEB9 E6    9F D052 A         LDB    [.CAAD]
05627A FEBD 03    1C       A        COM    .CURST   IS THE BEFORE STATE REVERSE?
05628A FEBF 27    02   FEC3         BEQ    CURTM5   YES, NOW IS THE NORMAL.
05629A FEC1 C8    08       A        EORB   #AMRV
05630A FEC3 BD    E51C     A CURTM5 JSR    CDSPC
05631A FEC6 86    14       A        LDA    #20
05632A FEC8 97    1B       A        STA    .CURTM
05633             FECA     A CURTM9 EQU    *
05634A FECA 108E  0000     A        LDY    #0
```

```
05635A FECE 8E    D00E     A        LDX
05636A FED1 CE    FF4E     A        LDU
05637A FED4 C6    04       A        LDB
05638A FED6 6C    82       A TM020  INC
05639A FED8 A6    84       A        LDA
05640A FEDA A1    C0       A        CMPA
05641A FEDC 25    0E   FEEC         BLO
05642A FEDE 6F    84       A        CLR
05643A FEE0 5A                      DECB
05644A FEE1 26    F3   FED6         BNE
05645A FEE3 96    09       A        LDA
05646A FEE5 85    08       A        BITA
05647A FEE7 27    03   FEEC         BEQ
05648A FEE9 31    A8 40    A        LEAY
05649                        *
05650A FEEC 96    09       A TM100  LDA
05651A FEEE 85    01       A        BITA
05652A FEF0 27    0F   FF01         BEQ
05653A FEF2 9E    0A       A        LDX
05654A FEF4 9C    0E       A        CMPX
05655A FEF6 26    09   FF01         BNE
05656A FEF8 9E    0C       A        LDX
05657A FEFA 9C    10       A        CMPX
05658A FEFC 26    03   FF01         BNE
05659A FEFE 31    A8 10    A        LEAY
05660                        *
05661A FF01 DC    14       A TM200  LDD
05662A FF03 C3    FFFF     A        ADDD
05663A FF06 DD    14       A        STD
05664A FF08 25    08   FF12         BCS
05665A FF0A 9E    12       A        LDX
05666A FF0C 30    1F       A        LEAX
05667A FF0E 9F    12       A        STX
05668A FF10 20    1F   FF31         BRA
05669A FF12 26    1D   FF31 TM220   BNE
05670A FF14 DC    12       A        LDD
05671A FF16 26    19   FF31         BNE
05672A FF18 DC    16       A        LDD
05673A FF1A DD    12       A        STD
05674A FF1C DC    18       A        LDD
05675A FF1E DD    14       A        STD
05676A FF20 96    09       A        LDA
05677A FF22 85    02       A        BITA
05678A FF24 27    0B   FF31         BEQ
05679A FF26 31    A8 20    A        LEAY
05680A FF29 85    04       A        BITA
05681A FF2B 27    04   FF31         BEQ
05682A FF2D 88    02       A        EORA
05683A FF2F 97    09       A        STA
05684                        *
05685             FF31     A TM900  EQU
05686A FF31 B6    D381     A        LDA
05687A FF34 D6    07       A        LDB
05688A FF36 27    04   FF3C         BEQ
05689A FF38 84    F0       A        ANDA
05690A FF3A 0F    07       A        CLR
05691A FF3C 34    20       A TM910  PSHS
05692A FF3E EA    61       A        ORB
05693A FF40 27    0A   FF4C         BEQ
05694A FF42 34    02       A        PSHS
05695A FF44 EA    E0       A        ORB
05696A FF46 F7    D381     A        STB
05697A FF49                         SYSREQ
     A FF49 B5    D404     A        BITA
05698A FF4C                  TM990  RTSPUL
     A FF4C 35    A0       A        PULS
05699                        *
05700A FF4E       32       A TMDAT  FCB
     A FF4F       3C       A
     A FF50       3C       A
     A FF51       18       A
05701                        *
05702             0001     A TMTCT  EQU
05703             0002     A TMTCI  EQU
05704             0004     A TMTCO  EQU
05705             0008     A NEWDAT EQU
05707                        *
05708                        * TEST COMMAND
05709                        *
05710                        * THIS COMMAND
05711                        * SUB-SYSTEM'S
05712                        * ADDRESSING.
05713                        *
05714                        * TEST SUB COM
05715                        *
05716                        * $90 RETURN
05717                        * $91 MOVE DAT
05718                        * $92 BRANCH T
05719                        * $93 BRANCH T
05720                        *
05721                        * OTHER CODE
05722                        *
05723             FF52     A TEST   EQU
05724A FF52 8E    D383     A        LDX
05725A FF55 CE    FFA0     A        LDU
05726A FF58 EC    81       A TSTKLP LDD
05727A FF5A 10A3  C1       A        CMPD
05728A FF5D 26    19   FF78         BNE
05729A FF5F 8C    D38B     A        CMPX
05730A FF62 25    F4   FF58         BLO
05731A FF64 A6    80       A TSTCMD LDA
05732A FF66 80    90       A        SUBA
05733A FF68 2B    0B   FF75         BMI
05734A FF6A 27    0C   FF78         BEQ
05735A FF6C 4A                      DECA
05736A FF6D 27    18   FF87         BEQ
05737A FF6F 4A                      DECA
```

```
#.$T1+4
#TMDAT
#4
,-X
,X
,U+
TM100
,X

TM020
.$TC
#NEWDAT
TM100
NWDI,Y

.$TC
#TMTCT  ENABLE?
TM200   NO. DISABLE
.$T1    YES.
.$T1I   SAME TIME?
TM200   NO.
.$T1+2
.$T1I+2
TM200   NO.
TMI,Y   YES. SET FLAG

.$T2+2  DECREMENT COUNTER
#$FFFF
.$T2+2
TM220
.$T2
-1,X
.$T2
TM900
TM900   NOT ZERO
.$T2
TM900   NOTZERO
.$T2D   ZERO
.$T2    RESTORE
.$T2D+2
.$T2+2
.$TC
#TMTCI  INTERVAL ENABLE?
TM900   DISABLE
INI,Y   YES. SET FLAG
#TMTCO  ONE SHOT?
TM900   NO.
#TMTCI  YES.
.$TC    INTERVAL OFF

*
STATUS
.PFIRQ  DID PF PUSH?
TM910   NO.
#PFI$   YES, CLEAR PF FIELD.
.PFIRQ
Y
1,S     OR TIMER FLAGS.
TM990   NO SYSREQ EXIST.
A
,S+
STATUS  SET INTERRUPT FLAGS.
*       SYSTEM REQUEST
ATTENT
Y
Y,PC

50,60,60,24



$01
$02
$04
$08


 IS ABLE TO REFER TO THE
 AREA IS ADDRESSED BY ABSOLUTE


IAND

) SYSTEM
)
) SUB COMMAND
) MACHINE LANGUAGE SUBROUTINE

:VE "INVALID COMMAND" ERROR.

:CMDPRM IS THIS KEYWORD RIGHT?
:TSTKEY
 X++
 U++
'STRTS  NO, COMMAND ENDED.
:CMDPRM+8 ALL EQUAL?
STKLP   NO, GODO MORE.
 X+     YES, WHAT IS SUBCMD.?
 $90
 STERR
 STRTS  RETURN

 STMOV  MOVE
```

```
05738A FF70 27    07   FF79     BEQ   TSTBRA   BRANCH
05739A FF72 4A                  DECA
05740A FF73 27    08   FF7D     BEQ   TSTBSR   CALL MACHINE
05741A FF75 7E    E0E6     A TSTERR JMP ERR70  INVALID COMMAND.
05742A FF78 39             TSTRTS RTS          RETURN TO SYSTEM
05743                           *
05744A FF79 AE    84       A TSTBRA LDX ,X     BRANCH ADDRESS LOAD.
05745A FF7B 20    E7   FF64     BRA   TSTCMD
05746                           *
05747A FF7D EE    81       A TSTBSR LDU ,X++   MACHINE ADDRESS LOADED.
05748A FF7F 34    10       A    PSHS  X        SAVE SUBCMD. PTR.
05749A FF81 AD    C4       A    JSR   ,U       SUBROUTINE CALL.
05750A FF83 35    10       A    PULS  X        RESTORE SUBCMD. PTR..
05751A FF85 20    DD   FF64     BRA   TSTCMD
05752A FF87 10AE  81       A TSTMOV LDY ,X++   LOAD SOURCE ADDRESS.
05753A FF8A EE    81       A    LDU   ,X++     LOAD DESTINATION ADDR..
05754A FF8C EC    81       A    LDD   ,X++     LOAD BYTE COUNT.
05755A FF8E 27    D4   FF64     BEQ   TSTCMD   NO MOVED DATA.
05756A FF90 34    10       A    PSHS  X        SAVE
05757A FF92 1F    01       A    TFR   D,X
05758A FF94 A6    A0       A TSTMLP LDA ,Y+
05759A FF96 A7    C0       A    STA   ,U+
05760A FF98 30    1F       A    LEAX  -1,X     FINISH?
05761A FF9A 26    F8   FF94     BNE   TSTMLP   NO, GO DO MORE.
05762A FF9C 35    10       A    PULS  X        RESTORE
05763A FF9E 20    C4   FF64     BRA   TSTCMD
05764                           *
05765A FFA0       59       A TSTKEY FCB     $59,$41,$4D,$41,$55,$43,$48,$49
     A FFA1       41       A
     A FFA2       4D       A
     A FFA3       41       A
     A FFA4       55       A
     A FFA5       43       A
     A FFA6       48       A
     A FFA7       49       A
```

```
                        NAM   HBIOS
                        OPT   LLE=120,REL,L
                        XREF  ABTCHK
                        XDEF  CBIOS
          0313   A ABORTF EQU $0313      ABORT FLAG.
                 *
P 0000                  PSCT             BIOS PROGRAM SECTION.
                 *
                 ***************************************
                 *
                 * FUJITSU PERSONAL COMPUTER
                 * BASIC INPUT/OUTPUT SYSTEM
                 * FOR F-BASIC
                 *
                 *   DESIGNED BY K.K
                 *   CODED    BY K.K
                 *   DATE     OCTOBER '80
                 ***************************************
                 *
                 ***************************************
                 *
                 * :: STRUCTURE
                 *
                 *   BIOS    EQU            BIOS ENTRY POINT
                 *            .
                 *            .
                 *           JSR    O,Y     EACH DRIVER
                 *                          PROCESSING
                 *            .
                 *            .
                 *           RTS            RETURN
                 *
                 ***************************************
                 ***************************************
                 *
                 * :: BIOS ROUTINE CALLING SEQUENCE
                 *
                 * 1. SET RCB CONTENTS
                 * 2. LDX #RCB ADDRESS
                 * 3. JSR [BIOS]
                 * 4.C (CARRY) CHECK
                 *
                 ***************************************
                 *
                 ***************************************
                 *
                 * :: RCB STRUCTURE
                 *
                 * LABEL    OFFSET       CONTENTS
                 *
          0000   A RQNO   EQU  0         REQUEST CODE
          0001   A RCBSTA EQU  1         STATUS
          0002   A RCBDBA EQU  2         BUFFER ADDRESS
          0002   A RCBCNL EQU  2         ANALOG PORT CHANNEL NUMBER.
          0002   A MOTORF EQU  2         CASSETTE MOTOR ON,OFF FLAG.
          0002   A CRWDAT EQU  2         CASSETTE READ/WRITE DATA.
                 *
          0003   A RCBRNG EQU  3         ANALOG PORT VOLTAGE RANGE.
                 *
          0004   A RCBCDT EQU  4         HARD COPY COLOR DATA.
          0004   A RCBLNH EQU  4         NUMBER OF BYTES (H)
          0004   A RCBTRK EQU  4         TRACK NUMBER
          0004   A RCBPAH EQU  4         PAGE ADDRESS    (H)
          0004   A RCBAPD EQU  4         ANALOG PORT INPUT DATA.
          0004   A RCBJCD EQU  4         KANJI JIS CODE.
                 *
          0005   A RCBLNL EQU  5         NUMBER OF BYTES (L)
          0005   A RCBSCT EQU  5         SECTOR NUMBER
          0005   A RCBPAL EQU  5         PAGE ADDRESS    (L)
                 *
          0006   A RCBSID EQU  6         DISK SIDE
          0006   A RCBPCH EQU  6         PAGE COUNTS HIGH
          0006   A RCBBMH EQU  6         MAX BUFFER LENGTH H
                 *
          0007   A RCBUNT EQU  7         DRIVE UNIT
          0007   A RCBPCL EQU  7         PAGE COUNTS LOW
          0007   A RCBBML EQU  7         MAX BUFFER LENGTH L
          0007   A RCBMTB EQU  7         SUBIN MAX TRANSFER BYTES.
                 *
                 *
                 * MOTORF : FF=ON,00=OFF
                 *
                 * :: RCBTRK STRUCTURE
                 *
                 * B7 - 0      TRACK NUMBER $00 - $27
                 *
                 *
                 * :: RCBSCT STRUCTURE
                 *
                 * B7 - 0      SECTOR NUMBER
                 *
                 *
                 * :: RCBPAH STRUCTURE
                 *
                 * B7(MSB)     BUBBLE CASSET NUMBER
                 * B6 - 0      PAGE ADDRESS   (H)
                 *
                 * :: RCBSID STRUCTURE
                 *
                 * B0 DISK SIDE   0 SIDE 0
                 *                1 SIDE 1
                 * B7 - B1          DON'T CARE
                 *
                 *
                 * :: RCBUNT
                 *
                 * B1 - B0  DRIVE UNIT NO.
                 * B7 - B2   DON'T CARE
```

```
                 *
                 *******
                 * ENTRY
                 * BIOS
                 *
                 * VECTOR
                 *
          FFFE   A PRESET
          FFFC   A PNMI
          FFFA   A PSWI
          FFF8   A PIRQ
          FFF6   A PFIRQ
          FFF4   A PSWI2
          FFF2   A PSWI3
          FFF0   A PVCTO1
          FFE6   A SINMAX
          FFE8   A DSTDRV
          FFE5   A DMOTOR
          FFE1   A CURTRK
          FFE0   A CRDRIV
                 *
                 * SHARED
                 *
          FC80   A SHARED
          0000   A SMSTA
          0001   A SMCFG
          0002   A SMCMD
          0003   A SMDAT
          0080   A SMEND
                 *
          000C   A FF
          000A   A LF
          001B   A ESC
          0029   A INKEY
          0003   A PUT
          0004   A GET
          0005   A GETC
          001B   A GETB1
                 *  ::
                 *
                 * $FD00
                 *   B7
                 *   B6
                 *   B5
                 *   B4
                 *   B1
                 *   B0
                 *
                 * $FD01
                 *   B7
                 *   B6
                 *   B5
                 *   B4
                 *   B3
                 *   B2
                 *   B1
                 *   B0
                 *
                 * $FD02
                 *   B7
                 *
                 *   B6
                 *   B5
                 *   B4
                 *   B3
                 *   B2
                 *   B1
                 *   B0
                 *
                 * $FD03
                 *   B7
                 *   B6
                 *   B5
                 *   B0
                 *
                 *******
                 *
                 * $FD04
                 *   B7
                 *  B2
                 *  B1
                 *  B0
                 *
                 * $FD05
                 *   B7
                 *   B6
                 *   B5
                 *   B0
                 *
                 *******
                 *
                 * $FD06
                 * D7-D0
                 *
                 * $FD07
                 * (READ
                 *
                 * B7 DS
                 * B6 SY
                 * B5 FR
                 * B4 OV
                 * B3 PA
                 * B2 TR
                 * B1 RE
                 * B0 TR
                 *
                 * (WRIT
                 *
```

```
***************************

 ADDRESS  $F000
AREA END  $FB7F

 ADDRESS EQU DEFINITION

EQU    $FFFE    RESET VECTOR
EQU    $FFFC    NMI   VECTOR
EQU    $FFFA    SWI   VECTOR
EQU    $FFF8    IRQ   VECTOR
EQU    $FFF6    FIRQ  VECTOR
EQU    $FFF4    SWI2  VECTOR
EQU    $FFF2    SWI3  VECTOR
EQU    $FFF0    NON USE
EQU    $FFE6    SUB WORK.MAX BUFFER CHECK.
EQU    $FFE8
EQU    $FFE5    MOTOR FLAG AREA
EQU    $FFE1    CURRENT TRACK ADDRESS.
EQU    $FFE0    CURRENT DRIVE.

 MEMORY EQUATE DEFINITION

EQU    $FC80
EQU    0
EQU    1
EQU    2
EQU    3
EQU    128

EQU    $0C      FORM FEED
EQU    $0A      LINEFEED
EQU    $1B      ESCAPE
EQU    $29      INKEY
EQU    $03      PUT
EQU    $04      GET
EQU    $05      GETC
EQU    27       GET BLOCK 1
I/O AREA MAP

   CASSET,LP OUTPUT
   SLCTIN (LP)
   STRB   (LP)
   RESERVE
 -2 NON CONECTION
AUDIO CASSETTE REMOTE CONTROL.
   MIC (CASSET)

   LP DATA OUTPUT
   DATA 7
   DATA 6
   DATA 5
   DATA 4
   DATA 3
   DATA 2
   DATA 1
   DATA 0

   CASSET,LP INPUT
   SPEAKER (CASSET)
   NOT IN USE
   DET2     (LP)
   DET1     (LP)
   PE       (LP)
   ACKNG    (LP)
   ERROR    (LP)
   BUSY     (LP)

   BELL OUTPUT
   BELL ON/OFF
   BELL (SHORT SOUND)
 -1 NON CONECTION
   SPEAKER SWITCH

********************************

   SUBSYSTEM INPUT
   BUSY
 PARITY ERROR
 STOP KEY INDICATOR
 ATTENSION

   SUBSYSTEM OUTPUT
   HALT
   IRQREQ (SUB: CANCEL)
 -1 NON CONECTION
   6809/Z80 FLAG.

********************************

 RS232C INTERFACE INPUT/OUTPUT
 SERIAL INPUT/OUTPUT DATA.

 RS232C INTERFACE STATUS/COMMAND.
  -- STATUS REGISTER

R
NDET
AME ERROR
ER RUN ERROR
RITY ERROR
ANSFER MISSING ERROR
CEIVE READY
ANSFER DATA BUFFER EMPTY

E) -- MODE COMMAND DATA
```

```
                  * B7 SCS
                  * B6 ESD
                  * B5 EP
                  * B4 PE
                  * B3-B2 L2,L1 CHARACTER LENGTH.
                  * B1-B0 B2,B1 BAUDE RATE.
                  * (WRITE) -- COMMAND DATA.
                  *
                  * B7 EH
                  * B6 IR
                  * B5 RTS
                  * B4 ER
                  * B3 SBRK
                  * B2 RXE
                  * B1 DTR
                  * B0 TXEN
                  *
                  *******************************************
                  *
                  * $FD08   VOICE GENERATION OUTPUT
                  *   B7-6 NON CONECTION
                  *   B5    COUNTER CLEAR
                  *   B4    NON CONECTION
                  *   B3    RESET
                  *   B2    INT
                  *   B1    STOP
                  *   B0    START
                  *
                  * $FD09   VOICE GENERATION OUTPUT
                  *
                  *   B7    CM3
                  *   B6    CM2
                  *   B5    CM1
                  *   B4    CM0
                  *   B3    DATA 3
                  *   B2    DATA 2
                  *   B1    DATA 1
                  *   B0    DATA 0
                  *
                  * $FD0A    VOICE GENERATION INPUT
                  *   B7    BUSY
                  *   B6-0 NON CONECTION
                  *
                  * $FD0B   VOICE DATA
                  *
                  *******************************************
                  * $FD0C A/D CONVERTER
                  * (READ)
                  * B7-B1 NOT IN USE.
                  * B0 ANALOG SERIAL DATA INPUT.
                  *
                  * (WRITE)
                  * B7-B4 NOT IN USE.
                  * B3 CHIP SELECT
                  * B2 REGISTER SELECT
                  * B1-B0 CHANNEL SELECT
                  *
                  * $FD0D - $FD0F NON USE
                  *******************************************
                  *
                  * $FD10   BUBBLE DATA REGISTER
                  * $FD11   BUBBLE COMMAND REGISTER
                  * $FD12   BUBBLE STATUS REGISTER
                  * $FD13   BUBBLE ERROR STATUS REGISTER
                  * $FD14   BUBBLE PAGE REGISTER (H)
                  * $FD15   BUBBLE PAGE REGISTER (L)
                  * $FD16   BUBBLE PAGE COUNT REGISTER (H)
                  * $FD17   BUBBLE PAGE COUNT REGISTER (L)
                  *
                  *******************************************
                  *
                  * $FD18   DISK STATUS REGISTER (READ)
                  *                COMMAND REGISTER (WRITE)
                  * $FD19   DISK TRACK REGISTER
                  * $FD1A   DISK SECTOR REGISTER
                  * $FD1B   DISK DATA REGISTER
                  * $FD1C   DISK SIDE REGISTER (B7)
                  * $FD1D   DISK DRIVE REGISTER
                  *         B7 MOTOR OFF : 0
                  *            MOTOR ON  : 1
                  *     B1-0 DRIVE NUMBER (0 - 3)
                  *
                  * $FD1E   NON USE
                  * $FD1F   DRQ DATA REQUEST (B7),IRQ REQUEST (B6)
                  *
                  *******************************************
                  * $FD20 KANJI ROM ADDRESS(H)
                  * $FD21 (L)
                  * $FD22 KANJI DATA (L)
                  * $FD23 (R)
                  *
                  * $FD24 - $FDFF NON USE
                  *
                  *******************************************
                  * I/O ADDRESS EQUATION
                  *
 0000   A MTLPOT EQU    $0000
 0000   A WRITE  EQU    MTLPOT
 0000   A REMOTE EQU    $0000
 0001   A LPDTOT EQU    $0001
 0002   A MTLPIN EQU    $0002
 0002   A READ   EQU    MTLPIN
 0003   A BELL   EQU    $0003

 0004   A STOPF  EQU    $0004
 0005   A SBSIN  EQU    $0005
```

```
                 0005     A SBSOUT EQU    $0005
                            *
                 0006     A RS232D EQU    $0006
                 0007     A RS232S EQU    $0007
                 0007     A RS232C EQU    $0007
                            *
                 0008     A VOICNT EQU    $0008
                 0009     A VOICMD EQU    $0009
                 000A     A VOICIN EQU    $000A
                 000B     A VOICDT EQU    $000B
                 000C     A ADCONV EQU    $000C
                            *
                 0010     A BDATA  EQU    $0010
                 0011     A BCMD   EQU    $0011
                 0012     A BSTAT  EQU    $0012
                 0013     A BERRST EQU    $0013
                 0014     A BPGADH EQU    $0014
                 0015     A BPGADL EQU    $0015
                 0016     A BPGCTH EQU    $0016
                 0017     A BPGCTL EQU    $0017
                            *
                            *
                 0018     A DSTAT  EQU    $0018
                 0018     A DCMD   EQU    $0018
                 0019     A DTRACK EQU    $0019
                 001A     A DSECTR EQU    $001A
                 001B     A DDATA  EQU    $001B
                 001C     A DSIDE  EQU    $001C
                 001D     A DDRIVE EQU    $001D
                 001E     A DITREQ EQU    $001E     INTERRUPU REQUEST
                 001F     A DDRQ   EQU    $001F     DATA REQUEST
                            *
                 0020     A KANROA EQU    $0020     KANJI ROM ADDRESS
                 0022     A KANROD EQU    $0022     KANJI ROM DATA
                 0021     A SKIP1  EQU    $21       BRN 1 BYTE SKIP
                 008E     A SKIP2  EQU    $8E       CMPX # 2 BYTES SKIP
                 1A50     A MSKFIQ EQU    $1A50     : ORCC #$50
                 000A     A FDREAD EQU    10        FLOPPY SECTOR READ.
                 0008     A FDREST EQU    8         FLOPPY RESTOR. SEEK TRACK 0.

                            *******************************************************
                            *
                            * *** BIOS MAIN PROGRAM ***
                            *
                            *  ** FUNCTION
                            *     1. JUMP TO EACH BIOS ROUTINE
                            *        ACCORDING TO REQUEST CODE
                            *     2. SET STATUS
                            *
                            *         IF ERROR  C=1 RCBSTA=ERROR NO.
                            *            ELSE   C=0 RCBSTA=0
                            *
                            *  ** INTERFACE
                            *
                            *     ENTRY - (X)=RCB ADDRESS
                            *
                            *     EXIT -   (X,Y,D,U,DP,CC(E,F,I ONLY))=PRESERVED
                            *        C=0  NON ERROR
                            *        C=1  ERROR DETECTED
                            *
                            *******************************************************
                 0000     P CBIOS  EQU    *         MAIN PROGRAM
0000 34    7F     A        PSHS   U,Y,X,DP,D,CC REGISTER SAVE
0002 86    FD     A        LDA    #$FD
0004 1F    8B     A        TFR    A,DP      DIRECT PAGE=$FD..
0006 E6    84     A        LDB    RQNO,X
0008 4F                    CLRA
0009 58                    ASLB             REQUEST NO. *2
000A 49                    ROLA
                            *
000B 10BE 002C    P        LDY    #BIOSTB
000F 31    AB     A        LEAY   D,Y
0011 86    02     A        LDA    #2        ASSUME RCB ERROR
0013 108C 0062    P        CMPY   #BTEND    REQUEST NO. CHECK
0017 22    06  001F        BHI    BIOS1     REQUEST NO. ERROR
                            *
0019 34    10     A        PSHS   X
001B AD    B4     A        JSR    [0,Y]     EACH PROCESS
001D 35    10     A        PULS   X         X,CC COME BACK
001F 35    01     A BIOS1  PULS   CC        CC COME BACK.
                            *
0021 A7    01     A        STA    RCBSTA,X ACCA=0 NORMAL,NOT 0 : ERROR
0023 26    03  0028        BNE    BIOS2
0025 1C    FE              CLC              CLEAR CARRY
0027       8E     A        FCB    SKIP2     CMPX # : SKIP NEXT 2 BYTES.
                            *
0028 1A    01       BIOS2  SEC              SET CARRY
002A 35    FE     A BIOS25 PULS   U,Y,X,DP,D,PC REGISTER COME BACK
           002C   P BIOSTB EQU    *
002C       0064   P        FDB    ANALGP    0 ANALOG PORT INPUT.
002E       0126   P        FDB    MOTOR     1 CASSETTE MOTOR CONTROLE.
0030       0133   P        FDB    CTBWRT    2 CASSETTE BYTE WRITE.
0032       01AC   P        FDB    CTBRED    3 CASSETTE BYTE READ.
0034       0239   P        FDB    INTBBL    4 INITIALIZE BUBBLE
0036       06CD   P        FDB    SCREEN    5 SCREEN COPY TO PRINTER.
0038       027F   P        FDB    WRTBBL    6 BUBBLE N PAGE WRITE
003A       027F   P        FDB    REDBBL    7 BUBBLE N PAGE READ
003C       FE02   A        FDB    RESTOR    8 SEEK TRACK 0
003E       FE05   A        FDB    DWRITE    9 1 SECTOR WRITE
0040       FE08   A        FDB    DREAD     10 1 SECTOR READ
0042       0899   P        FDB    VOICE     11 VOICE   OUTPUT
0044       02F7   P        FDB    BEEPON    12 BELL ON
0046       02FA   P        FDB    BEEPOF    13 BELL OFF
0048       0300   P        FDB    LPOUT     14 PRINTER OUTPUT
004A       050C   P        FDB    HDCOPY    15 CRT HARD COPY
004C       0347   P        FDB    SUBOUT    16 SUBSYSTEM OUTPUT
004E       0347   P        FDB
0050       03AB   P        FDB
0052       0469   P        FDB
0054       0483   P        FDB
0056       04E3   P        FDB
0058       00A2   P        FDB
005A       031D   P        FDB
005C       08C0   P        FDB
005E       082D   P        FDB
0060       0870   P        FDB
0062       08AF   P BTEND  FDB
                            *
                            ***************
                            *
           0064   P ANALGP EQU
0064 A6    02     A        LDA
0066 84    03     A        ANDA
0068 34    02     A        PSHS
006A E6    03     A        LDB
006C 4F                    CLRA
006D C1    48     A        CMPB
006F 26    02  0073        BNE
0071 86    04     A        LDA
0073 AA    E4     A RSSET  ORA
0075 A7    E4     A        STA
0077 8A    08     A        ORA
0079 97    0C     A        STA
007B 84    07     A        ANDA
007D 97    0C     A        STA
007F 34    02     A        PSHS
0081 86    09     A        LDA
0083 6F    E4     A        CLR
0085 68    E4     A SDREAD ASL
0087 C6    80     A        LDB
0089 EA    61     A        ORB
008B D7    0C     A        STB
008D C4    07     A        ANDB
008F D7    0C     A        STB
0091 D6    0C     A        LDB
0093 C4    01     A        ANDB
0095 EB    E4     A        ADDB
0097 E7    E4     A        STB
0099 4A                    DECA
009A 26    E9  0085        BNE
009C 35    06     A        PULS
009E A7    04     A        STA
00A0 4F                    CLRA
00A1 39                    RTS
                            ***************
                            *
           00A2   P KANJIR EQU
00A2 EE    02     A        LDU
00A4 108E 0000    A        LDY
00A8 C6    10     A        LDB
00AA 10AF C1      A BFCLR  STY
00AD 5A                    DECB
00AE 26    FA  00AA        BNE
00B0 C6    10     A        LDB
00B2 34    04     A        PSHS
00B4 EC    04     A        LDD
00B6 8D    1C  00D4        BSR
00B8 25    17  00D1        BCS
00BA EE    02     A        LDU
00BC DD    20     A KANEXD STD
00BE 109E 22      A        LDY
00C1 108C 0001    A        CMPY
00C5 27    0A  00D1        BEQ
00C7 10AF C1      A        STY
00CA C3    0001   A        ADDD
00CD 6A    E4     A        DEC
00CF 26    EB  00BC        BNE
00D1 4F                KANRTS CLRA
00D2 35    84     A        PULS
                            *
                            * 'CNVJTK'
                            * CONVERT THE
                            * INPUT  D--K
                            *
                            * OUTPUT D--R
                            *
                            * OTHER REGIS
                            *
                            * LOGIC
                            * REG JIS(15:
                            * REG A(15:0)
                            * REG T(2:0);
                            * A<-0;
                            * A(8:4)<-JIS
                            * A(12:9)<-JI
                            * T<-(JIS(14:
                            * IF T>2 THEN
                            * ELSE IF T=2
                            * T<-0;
                            * A(12)<-1;EN
                            * A(15:13)<-T
                            *
00D4 84    7F     A CNVJTK ANDA
00D6 81    21     A        CMPA
00D8 25    10  00EA        BLO
00DA 81    28     A        CMPA
00DC 25    08  00E6        BLO
00DE 81    30     A        CMPA
00E0 25    08  00EA        BLO
00E2 81    50     A        CMPA
00E4 24    04  00EA        BHS
00E6 C5    60     A SCDCK  BITB
00E8 26    03  00ED        BNE
```

```
UBIN     17 SUBSYSTEM INPUT
NPUT     18 CONSOLE INPUT
NPUTC    19 CONSOLE INPUT CONTINUE
UTPUT    20 CONSOLE OUTPUT
EYIN     21 KEY INPUT
ANJIR    22 KANJI ROM DATA READ.
PCHK     23 LP READY CHECK
IINIT    24 BIOS INITIAL SET
PARM     25 VOICE PARAMETER SET
DPUT     26 VOICE DATA TRANSFER
RESET    27 VOICE DEVICE RESET

*************************

         ANALOG PORT INPUT ROUTINE.
CBCNL,X GET CHANNEL NUMBER.
$03      EXTRACT C1,CO

CBRNG,X GET VOLTAGE RANGE.

H        HIGH VOLTAGE RANGE?
SSET     NO
$04      RS=1 SET
S
S
$08
DCONV
$07
DCONV

9        COUNTER=9 SET.
S        WORK AREA=0 CLEAR
S        RESULT SHIFT LEFT 1 BIT.
$80
S
DCONV
$07
DCONV
DCONV    READ SERIAL DATA(LSB)
$01      EXTRACT SERIAL 1 BIT DATA.
S        THIS TIME DATA ADD.
S        RESULT STORE.
         COUNT DOWN.
READ
         GET RESULT ONE BYTE.
BAPD,X REAULT STORE
         NORMAL RETURN

****************************

         KANJI ROM READ ROUTINE
BDBA,X GET BUFFER ADDRESS

6        SET COUNTER
U++
         32 BYTE 0 CLEAR
CLR
6
         LOOP COUNTER SET.
BJCD,X GET THE KANJI JIS CODE.
VJTK     CONVERT JIS CODE TO ROM ADDRESS
NRTS     NOT ON THE ROM.
BDBA,X GET KANJI DUFFER ADDRESS.
NROA     ADDRESS SET TO ROM ADDRESS REGISTER.
NROD     GET KANJI DATA
         NULL DATA ?
NRTS     NON DATA.
U++
         ADDRESS COUNT UP
         DECREMENT COUNTER BY 1.
NEXD
         NORMAL RETURN
PC

S-KANJI CODE TO THE ROM ADDRESS.
I CODE.

ADDRESS.

 ARE RESERVED.

0);
1:8);
-2)*3+JIS(6:5)-1;
-T-1;
EN DO:

7F       IS THE CODE ON THE KANJI ROM ?
21
ROM      NO.
28
DCK
30
ROM      NO.
50
ROM      NO.
60
ROM      YES.
```

```
00546P 00EA 1A    01             NOROM   SEC
00547P 00EC 39                           RTS
00548                                    *
00549P 00ED 34    06           A ONROM   PSHS   D          SAVE THE JIS.
00550P 00EF 84    0F           A         ANDA   #$0F       PICK UP JIS(11:8)
00551P 00F1 C4    1F           A         ANDB   #$1F       PICKUP JIS(4:0)
00552P 00F3 58                           LSLB              SET IT TO  A(12:4)
00553P 00F4 58                           LSLB              AND OTHER FIELEDS IS SET 0.
00554P 00F5 58                           LSLB
00555P 00F6 58                           LSLB
00556P 00F7 49                           ROLA
00557P 00F8 34    06           A         PSHS   D          SAVE THE ADDRESS.
00558P 00FA A6    62           A         LDA    2,S        PICKUP JIS(14:12)
00559P 00FC 84    70           A         ANDA   #$70       AND SUBTRUCT 2 FROM IT.
00560P 00FE 80    20           A         SUBA   #2*16
00561P 0100 34    02           A         PSHS   A          MULTIPLY IT BY 3.
00562P 0102 48                           LSLA
00563P 0103 AB    E4           A         ADDA   ,S
00564P 0105 48                           LSLA              SET IT TO THE HIGHEST
00565P 0106 A7    E4           A         STA    ,S         3 BITS AND SAVE.
00566P 0108 A6    64           A         LDA    4,S        PICKUP JIS(6:5) AND
00567P 010A 84    60           A         ANDA   #$60       SUBTRUCT 1.
00568P 010C 80    20           A         SUBA   #$20
00569P 010E AB    E4           A         ADDA   ,S         MAKE THE T.
00570P 0110 25    09    011B             BCS    CNVJHI
00571P 0112 81    40           A         CMPA   #2*32      IS THE T .LT. 2 ?
00572P 0114 25    07    011D             BLO    CNVJLO     YES.
00573P 0116 22    03    011B             BHI    CNVJHI     NO,IT IS .GT.2
00574P 0118 86    10           A         LDA    #$10       NO,IT IS 2. SET A(12) BIT
00575                            * SKIP2 CMPX #
00576P 011A       8C           A         FCB    $8C
00577P 011B 80    20           A CNVJHI  SUBA   #$1*32     SUBTRUCT 1 FROM THE T.
00578P 011D AA    61           A CNVJLO  ORA    1,S        SET THE T TO A(15:13).
00579P 011F E6    62           A         LDB    2,S        LOAD THE ADDRESS TO D-REG.
00580P 0121 32    65           A         LEAS   5,S
00581P 0123 1C    FE                     CLC
00582P 0125 39                           RTS
00584             00C2         A MOTRON  EQU    $C2
00585             00C0         A MOTROF  EQU    $C0
00586                            ********************************
00587                            *
00588             0126         P MOTOR   EQU    *          CASSETTE MOTOR CONTROL.
00589P 0126 A6    02           A         LDA    MOTORF,X   GET MOTOR FLAG.
00590P 0128 C6    C2           A         LDB    #MOTRON    ASSUME MOTOR ON.
00591P 012A 43                           COMA              MOTOR ON?
00592P 012B 27    02    012F             BEQ    MOTRCN     YES,MOTOR ON.
00593P 012D C6    C0           A         LDB    #MOTROF    NO,MOTOR OFF.
00594P 012F D7    00           A MOTRCN  STB    REMOTE     MOTOR VONTROL SET.
00595P 0131 4F                           CLRA              NORMAL RETURN.
00596P 0132 39                           RTS
00597                            *
00599                            ****************************************
00600                            *
00601             0133         P CTBWRT  EQU    *
00602P 0133 A6    02           A         LDA    CRWDAT,X   GET WRITE DATA.
00603P 0135 8D    63    019A             BSR    DOOUT
00604P 0137 8E    0006         A         LDX    #6         TIMER SET.
00605P 013A 8D    68    01A4             BSR    TIMER
00606P 013C 8D    61.   019F W20         BSR    D1OUT
00607P 013E 8E    0012         A         LDX    #18        TIMER SET.
00608P 0141 8D    61    01A4             BSR    TIMER
00609P 0143 8D    55    019A W30         BSR    DOOUT
00610P 0145 8E    0012         A         LDX    #18        TIMER SET.
00611P 0148 8D    5A    01A4             BSR    TIMER
00612P 014A 108E  0000         A         LDY    #0         COUNTER CLEAR.
00613P 014E 8D    4F    019F W40         BSR    D1OUT
00614P 0150 44                           LSRA              IS WRITING DATA 0?,OR 1?
00615P 0151 25    0E    0161             BCS    WRITE1     IF IT IS 1,GO TO WRITE1.
00616P 0153 8E    0011         A WRITE0  LDX    #17        TIMER SET.
00617P 0156 8D    4C    01A4             BSR    TIMER
00618P 0158 8D    40    019A W50         BSR    DOOUT
00619P 015A 8E    0011         A         LDX    #17        TIMER SET.
00620P 015D 8D    45    01A4             BSR    TIMER
00621P 015F 20    0E    016F             BRA    NEXT
00622P 0161 8E    0024         A WRITE1  LDX    #36        TIMER SET.
00623P 0164 8D    3E    01A4             BSR    TIMER
00624P 0166 8D    32    019A W60         BSR    DOOUT
00625P 0168 8E    0023         A         LDX    #35        TIMER SET.
00626P 016B 8D    37    01A4             BSR    TIMER
00627P 016D 20    00    016F             BRA    NEXT
00628P 016F 31    21           A NEXT    LEAY   1,Y        BIT COUNTS.
00629P 0171 108C  0008         A         CMPY   #8         1 BYTE END ?
00630P 0175 26    D7    014E             BNE    W40        IF NOT,GOTO NEXT BIT.
00631P 0177 108E  0000         A         LDY    #0         COUNTER CLEAR.
00632P 017B 8D    22    019F W70         BSR    D1OUT
00633P 017D 8E    0024         A         LDX    #36        TIMER SET.
00634P 0180 8D    22    01A4             BSR    TIMER
00635P 0182 8D    16    019A W80         BSR    DOOUT
00636P 0184 8E    000C         A         LDX    #12        TIMER SET.
00637P 0187 8D    1B    01A4             BSR    TIMER
00638P 0189 31    21           A         LEAY   1,Y        COUNTER INC.
00639P 018B 108C  0002         A         CMPY   #2         IS 2 STOP BITS WRITTEN ?
00640P 018F 27    07    0198             BEQ    BYTEND     IF YES,GO TO 1 BYTE END.
00641P 0191 8E    0016         A         LDX    #22        TIMER SET.
00642P 0194 8D    0E    01A4             BSR    TIMER
00643P 0196 20    E3    017B             BRA    W70        GO TO 2ND STOP BIT.
00644P 0198 4F                   BYTEND  CLRA
00645P 0199 39                           RTS               NOMAL RETURN.
00646                            *
00647P 019A C6    C2           A DOOUT   LDB    #$C2       OUTPUT LOW-LEVEL.
00648P 019C D7    00           A         STB    WRITE      DATA OUT.
00649P 019E 39                           RTS
00650                            *
00651P 019F C6    C3           A D1OUT   LDB    #$C3       OUTPUT HIGH-LEVEL.
00652P 01A1 D7    00           A         STB    WRITE      DATA OUT.
00653P 01A3 39                           RTS
00654                            *
```

```
00655P 01A4 30    1F        A TIMER  LEAX  -1,X
00656P 01A6 8C    0000      A        CMPX  #0
00657P 01A9 26    F9   01A4          BNE   TIMER
00658P 01AB 39                       RTS
00659                        *
00661                        ****************************************************
00662                        *
00663             01AC      P CTBRED EQU   *          CASSETTE BYTE READ.
00664P 01AC 34    10        A        PSHS  X          RCB ADDR SAVE.
00665P 01AE 108E  0000      A CBINO  LDY   #0         COUNTER CLEAR.
00666P 01B2 7D    0313      A R10    TST   ABORTF     STOP KEY ON ?
00667P 01B5 1026  FE47      A        LBNE  ABTCHK
00668P 01B9 8D    5A   0215          BSR   DIN
00669P 01BB 26    F5   01B2          BNE   R10        IF NOT,READ AGAIN.
00670P 01BD 8D    62   0221 R20      BSR   EDGE       EDGE CATCH.
00671P 01BF 8E    001E      A        LDX   #30        TIMER SET.
00672P 01C2 8D    E0   01A4          BSR   TIMER
00673P 01C4 8D    4F   0215 R30      BSR   DIN
00674P 01C6 26    EA   01B2          BNE   R10        IF IT IS NOT CAUGHT,START BIT AGAIN.
00675P 01C8 4F                       CLRA
00676P 01C9 8D    56   0221 R40      BSR   EDGE       EDGE CATCH.
00677P 01CB 8E    0019      A        LDX   #25        TIMER SET.
00678P 01CE 8D    D4   01A4          BSR   TIMER
00679P 01D0 8D    43   0215 R50      BSR   DIN
00680P 01D2 58                       LSLB
00681P 01D3 46                       RORA             SET DATA IN A-REG.
00682P 01D4 31    21        A        LEAY  1,Y        COUNTER INC.
00683P 01D6 108C  0008      A        CMPY  #8         ARE DATA 8 BIT ?
00684P 01DA 27    0F   01EB          BEQ   STOP       IF OK,GO TO STOP BIT CATCH.
00685P 01DC 7D    0313      A R60    TST   ABORTF     STOP KEY ON ?
00686P 01DF 1026  FE1D      A        LBNE  ABTCHK
00687P 01E3 D6    02        A        LDB   READ       DATA IN.
00688P 01E5 C4    80        A        ANDB  #$80
00689P 01E7 26    F3   01DC          BNE   R60        IF NOT,READ AGAIN.
00690P 01E9 20    DE   01C9          BRA   R40        GO TO NEXT BIT READ.
00691                        *
00692P 01EB 108E  0000      A STOP   LDY   #0         COUNTER CLEAR.
00693P 01EF 7D    0313      A R70    TST   ABORTF     STOP KEY ON ?
00694P 01F2 1026  FE0A      A        LBNE  ABTCHK
00695P 01F6 D6    02        A        LDB   READ       DATA IN.
00696P 01F8 C4    80        A        ANDB  #$80
00697P 01FA 26    F3   01EF          BNE   R70        IF NOT,READ AGAIN.
00698P 01FC 8D    23   0221 R80      BSR   EDGE       EDGE CATCH.
00699P 01FE 8E    0018      A        LDX   #24
00700P 0201 8D    A1   01A4          BSR   TIMER
00701P 0203 8D    10   0215 R90      BSR   DIN
00702P 0205 27    A7   01AE          BEQ   CBINO
00703P 0207 31    21        A        LEAY  1,Y        COUNTER INC.
00704P 0209 108C  0002      A        CMPY  #2         ARE 2 STOP BIT CAUGHT ?
00705P 020D 26    E0   01EF          BNE   R70        GO TO 2ND STOP BIT.
00706P 020F 35    10        A        PULS  X          RCB COME BACK.
00707P 0211 A7    02        A        STA   CRWDAT,X
00708P 0213 4F                       CLRA             NOMAL READ.
00709P 0214 39                       RTS              1 BYTE DATA HAVE BEEN CAUGHT.
00710                        *
00711P 0215 D6    02        A DIN    LDB   READ       DATA IN.
00712P 0217 C4    80        A        ANDB  #$80
00713P 0219 39                       RTS
00714                        *
00715P 021A 35    10        A ERROR  PULS  X
00716P 021C 6F    02        A        CLR   CRWDAT,X
00717P 021E 86    34        A        LDA   #52        ABNOMAL READ.
00718P 0220 39                       RTS
00719                        *
00720P 0221 7D    0313      A EDGE   TST   ABORTF     STOP KEY ON ?
00721P 0224 1026  FDD8      A        LBNE  ABTCHK
00722P 0228 D6    02        A        LDB   READ       DATA IN.
00723P 022A C4    80        A        ANDB  #$80
00724P 022C C1    80        A        CMPB  #$80       RAISING EDGE OK ?
00725P 022E 26    F1   0221          BNE   EDGE       IF NOT,READ AGAIN.
00726P 0230 D6    02        A        LDB   READ       DATA IN.
00727P 0232 C4    80        A        ANDB  #$80
00728P 0234 C1    80        A        CMPB  #$80       RAISING EDGE OK ?
00729P 0236 26    E9   0221          BNE   EDGE       IF NOT,READ AGAIN.
00730P 0238 39                       RTS
00731                        *
00733                        ****************************************************
00734                        *
00735             0239      P INTBBL EQU   *          BUBBLE INITIALIZE
00736P 0239       1A50      A        FDB   MSKFIQ
00737P 023B 8D    30   026D          BSR   BUBRDY
00738P 023D 86    04        A        LDA   #$04       INITIALIZE BUBBLE COMMAND SET
00739P 023F 97    11        A INTB05 STA   BCMD
00740P 0241 96    12        A INTB10 LDA   BSTAT      CME ?
00741P 0243 2A    FC   0241          BPL   INTB10     NO,LOOP
00742P 0245 96    12        A        LDA   BSTAT      CME ?
00743P 0247 2A    FB   0241          BPL   INTB10     NO,LOOP
00744P 0249 85    02        A        BITA  #$02
00745P 024B 26    02   024F          BNE   BERROR     ERROR ANALYZE
00746P 024D 4F                       CLRA
00747P 024E 39                       RTS
00748                        *
00749             024F      P BERROR EQU   *          ERROR ANALYZE
00750P 024F D6    13        A        LDB   BERRST     LOAD ERROR STATUS REGISTER
00751P 0251 86    1E        A        LDA   #30        30
00752P 0253 54                       LSRB             UNDEFINED ?
00753P 0254 25    16   026C          BCS   BERR50
00754P 0256 4C                       INCA             31
00755P 0257 54                       LSRB             NO MARKER ?
00756P 0258 25    12   026C          BCS   BERR50
00757P 025A 4C                       INCA             32
00758P 025B 54                       LSRB             MANY BAD LOOP ?
00759P 025C 25    0E   026C          BCS   BERR50
00760P 025E 4C                       INCA             33
00761P 025F 54                       LSRB             TRANSFER MISSING ?
00762P 0260 25    0A   026C          BCS   BERR50
00763P 0262 4C                       INCA             34
```

```
00764P 0263 54                       LSRB
00765P 0264 25    06   026C          BCS
00766P 0266 4C                       INCA
00767P 0267 54                       LSRB
00768P 0268 25    02   026C          BCS
00769P 026A 86    25        A NONBUB LDA
00770P 026C 39                BERR50 RTS
00771                        *
00772             026D      P BUBRDY EQU
00773P 026D 108E  FFFF      A        LDY
00774P 0271 31    3F        A BUBLOP LEAY
00775P 0273 27    06   027B          BEQ
00776P 0275 96    12        A        LDA
00777P 0277 44                       LSRA
00778P 0278 25    F7   0271          BCS
00779P 027A 39                       RTS
00780
00781P 027B 32    62        A BUBERR LEAS
00782P 027D 20    EB   026A          BRA
00784             0007      A REDB   EQU
00785                        ************
00786                        *
00787             027F      P REDBBL EQU
00788P 027F       1A50      A WRTBBL FDB
00789P 0281 8D    EA   026D          BSR
00790P 0283 EC    06        A        LDD
00791P 0285 DD    16        A        STD
00792P 0287 EC    04        A        LDD
00793P 0289 8D    27   02B2          BSR
00794P 028B DD    14        A        STD
00795P 028D 10AE  02        A        LDY
00796P 0290 A6    84        A        LDA
00797P 0292 81    07        A        CMPA
00798P 0294 27    49   02DF          BEQ
00799P 0296 86    02        A        LDA
00800P 0298 97    11        A        STA
00801P 029A 96    12        A WRTB10 LDA
00802P 029C 2B    0A   02A8          BMI
00803P 029E 85    40        A        BITA
00804P 02A0 27    FB   029A          BEQ
00805P 02A2 A6    A0        A        LDA
00806P 02A4 97    10        A        STA
00807P 02A6 20    F2   029A          BRA
00808                        *
00809P 02A8 96    12        A WRTB20 LDA
00810P 02AA 2A    EE   029A          BPL
00811P 02AC 85    02        A WRTB30 BITA
00812P 02AE 26    9F   024F          BNE
00813P 02B0 4F                       CLRA
00814P 02B1 39                       RTS
00815                        *
00816             02B2      P REALAD EQU
00817                        *
00818P 02B2 34    06        A        PSHS
00819P 02B4 84    7F        A        ANDA
00820P 02B6 1083  03FF      A        CMPD
00821P 02BA 22    1E   02DA          BHI
00822                        *
00823P 02BC EE    06        A        LDU
00824P 02BE 1183  0400      A        CMPU
00825P 02C2 22    16   02DA          BHI
00826                        *
00827P 02C4 34    40        A        PSHS
00828P 02C6 E3    E1        A        ADDD
00829P 02C8 1083  0400      A        CMPD
00830P 02CC 22    0C   02DA          BHI
00831P 02CE 35    06        A        PULS
00832P 02D0 85    80        A        BITA
00833P 02D2 27    05   02D9          BEQ
00834P 02D4 84    7F        A        ANDA
00835P 02D6 C3    0400      A        ADDD
00836P 02D9 39                RADRTS RTS
00837                        *
00838P 02DA 86    23        A ADRERR LDA
00839P 02DC 35    60        A        PULS
00840                        Y:
00841P 02DE 39                       RTS
00843                        ************
00844                        *
00845             02DF      P REDBBX EQU
00846P 02DF 86    01        A        LDA
00847P 02E1 97    11        A        STA
00848P 02E3 96    12        A REDB10 LDA
00849P 02E5 2B    0A   02F1          BMI
00850P 02E7 85    20        A        BITA
00851P 02E9 27    FB   02E3          BEQ
00852P 02EB 96    10        A        LDA
00853P 02ED A7    A0        A        STA
00854P 02EF 20    F2   02E3          BRA
00855                        *
00856             02F1      P REDB20 EQU
00857P 02F1 96    12        A        LDA
00858P 02F3 2A    EE   02E3          BPL
00859P 02F5 20    B5   02AC          BRA
00860                        *
00862                        *
00863                        ************
00864                        *
00865             02F7      P BEEPON EQU
00866P 02F7 86    81        A        LDA
00867                        * SKIP 2 CMF
00868P 02F9       8C        A        FCB
00869                        *
00870             02FA      P BEEPOF EQU
00871P 02FA 86    01        A        LDA
00872P 02FC 97    03        A        STA
00873P 02FE 4F                BEEP50 CLRA
```

```
          CRC ERROR ?
BERR50
          35
          PAGE ADDRESS ERROR ?
BERR50
#37       BUBBLE DEVICE NOT READY.


*
#$FFFF
-1,Y
BUBERR
BSTAT

BUBLOP


2,S       STACK ADJUSTMENT.
NONBUB
?
******************************

*
MSKFIQ    IRQ MASK SEI.
BUBRDY
RCBPCH,X
BPGCTH
RCBPAH,X
REALAD    REAL ADDRESS SET. ADDRESS CHECK.
BPGADH
RCBDBA,X BUFFER ADDRESS
RQNO,X
#REDB     REDBBL ?
REDBBX    BRANCH BUBBLE READ.,
#$02      COMMAND SET
BCMD
BSTAT
WRTB20    COMMAND END
#$40
WRTB10    TDRA
0,Y+
BDATA
WRTB10

BSTAT
WRTB10
#$02      ERROR ?
BERROR


*         ADDRESS,COUNTS CHECK.
  UNIT OFFSET ADD.
D         D-SAVE.
#$7F      B15 -> 0 D:CASSETE ADDRESS.
#$3FF     ADDR <= $3FF ?
ADRERR    NO,PAGE ADDRESS ERROR.

RCBPCH,X
#$400     COUNTS <= $400 ?
ADRERR    NO,PAGE COUNTS ERROR.

U
,S++      ++: STACK ADJUST.
#$400     ADDR+COUNTS < $400 ?
ADRERR    NO,PAGE ADDRESS ERROR.
D         D-COME BACK.
#$80      UNIT 0 ?
RADRTS    YES,UNIT 0.
#$7F      ACCA: MSB=0
#$400     NO,UNIT1.  ADD UNITO OFFSET.

#35       PAGE ADDRESS ERROR.
U,Y       U:DUMMY FOR STACK ADJUST.
 "      FOR BSR REALAD.

**************************

*         BUBBLE READ
#$01      COMMAND SET
BCMD
BSTAT
REDB20    COMMAND END
#$20
REDB10    RDA
BDATA
0,Y+
REDB10

*
BSTAT
REDB10
WRTB30

***************************

*         BELL ON
#$81
#

$8C

*         BELL OFF
#$01
BELL
          NON ERROR
```

```
04375 00874P 02FF 39                     RTS
04385 00876                        ****************************************
04390 00877                        *
04395 00878            0300      P LPOUT  EQU   *          PRINTER OUTPUT
04400 00879P 0300 10AE 02        A        LDY   RCBDBA,X DATA BUFFER ADDRESS
04405 00880P 0303 EE   04        A        LDU   RCBLNH,X OUTPUT BYTES NO.
04410 00881P 0305 27   14   031B          BEQ   LPOUT2   IF BYTES NO.=0 DO NOTHING
04415 00882            0307      P LPOUT1 EQU   *
04420 00883P 0307 A6   A0        A        LDA   0,Y+
04425 00884P 0309 BD   065D      P        JSR   LPPUTX
04430 00885P 030C 7D   0313      A        TST   ABORTF
04435 00886P 030F 1026 FCED      A        LBNE  ABTCHK
04440 00887P 0313 33   5F        A        LEAU  -1,U       ALL END ?
04445 00888P 0315 1183 0000      A        CMPU  #0
04450 00889P 0319 26   EC   0307          BNE   LPOUT1
04455 00890P 031B 4F               LPOUT2 CLRA
04460 00891P 031C 39                      RTS
04465 00892                        ***********************************************************
04470 00893                        *
04475 00894            031D      P LPCHK  EQU   *          PRINTER OUTPUT
04480 00895P 031D D6   02        A        LDB   MTLPIN     ERROR ?
04485 00896P 031F C5   02        A        BITB  #$02
04490 00897P 0321 27   1B   033E          BEQ   LPCH40     YES ERROR
04495 00898                        *
04500 00899P 0323 CE   0002      A        LDU   #2         ABOUT 2.0 SECOUND
04505 00900P 0326 10BE F000      A LPCH10 LDY   #$F000
04510 00901P 032A 31   3F        A LPCH20 LEAY  -1,Y
04515 00902P 032C 27   07   0335          BEQ   LPCH30
04520 00903P 032E D6   02        A        LDB   MTLPIN
04525 00904P 0330 54                      LSRB             BUSY ?
04530 00905P 0331 25   F7   032A          BCS   LPCH20
04535 00906                        *
04540 00907P 0333 4F                      CLRA
04545 00908P 0334 39                      RTS
04550 00909                        *
04555 00910P 0335 33   5F        A LPCH30 LEAU  -1,U
04560 00911P 0337 1183 0000      A        CMPU  #0
04565 00912P 033B 26   E9   0326          BNE   LPCH10
04570 00913P 033D 5F                      CLRB
04575 00914                        *
04580 00915P 033E 86   32        A LPCH40 LDA   #50        PAPER EMPTY ?
04585 00916P 0340 C5   08        A        BITB  #$08
04590 00917P 0342 26   02   0346          BNE   LPCH50     YES PE
04595 00918P 0344 86   33        A        LDA   #51        NOT READY,HARD ERROR
04600 00919P 0346 39               LPCH50 RTS
04610 00921                        ***********************************************************
04615 00922                        *
04620 00923            0347      P SUBOUT EQU   *          SUBSYSTEM OUTPUT
04625 00924            0347      P SUBIN  EQU   *          SUBSYSTEM INPUT
04630 00925P 0347 86   01        A        LDA   #1         RCB ERROR SET
04635 00926P 0349 6D   04        A        TST   RCBLNH,X RCBLNH MUST BE 0.
04640 00927P 034B 26   5D   03AA          BNE   SUB70
04645 00928P 034D E6   05        A        LDB   RCBLNL,X BYTES NO (1 - 128)
04650 00929P 034F 27   59   03AA          BEQ   SUB70      IF BYTES LENGTH=0,ERROR
04655 00930P 0351 C1   80        A        CMPB  #128       TOO LONG ?
04660 00931P 0353 22   55   03AA          BHI   SUB70      YES RCB ERROR
04665 00932                        *
04670 00933P 0355 34   10        A        PSHS  X          ACCX SAVE RCB ADDR
04675 00934P 0357 108E FC80      A        LDY   #SHARED    DESTINATION
04680 00935P 035B AE   02        A        LDX   RCBDBA,X SOURCE
04685 00936P 035D BD   05EA      P        JSR   BYHL       BUSY CHECK AND HALT SET
04690 00937                        *
04695 00938P 0360 A6   01        A        LDA   1,X
04700 00939P 0362 84   80        A        ANDA  #$80
04705 00940P 0364 A7   01        A        STA   1,X
04710 00941P 0366 A6   21        A        LDA   1,Y        STATUS READ
04715 00942P 0368 84   7F        A        ANDA  #$7F       STATUS RESERVE
04720 00943P 036A AA   01        A        ORA   1,X        CONT FLAG SET
04725 00944P 036C A7   01        A        STA   1,X
04730 00945                        *
04735 00946P 036E A6   80        A SUB20  LDA   0,X+       OUT DATA READ
04740 00947P 0370 A7   A0        A SUB25  STA   0,Y+       DATA STORE
04745 00948P 0372 5A                      DECB
04750 00949P 0373 26   F9   036E          BNE   SUB20
04755 00950                        *
04760 00951P 0375 BD   0601      P SUB30  JSR   HRESET     HALT RESET COMMAND EXEC
04765 00952P 0378 35   10        A        PULS  X          ACCX COME BACK (RCB ADDRESS)
04770 00953P 037A 4F                      CLRA             ASSUME NORMAL RETURN.
04775 00954P 037B E6   84        A        LDB   0,X        READ REQUEST NO.
04780 00955P 037D C1   11        A        CMPB  #17        SUBIN ?
04785 00956P 037F 26   29   03AA          BNE   SUB70      NO (SUBOUT)
04790 00957                        *
04795 00958P 0381 BD   05E5      P        JSR   BYHLWD
04800 00959P 0384 BD   0619      P        JSR   READST     LOAD STATUS.
04805 00960P 0387 26   1B   03A4          BNE   SUB60      ERROR
04810 00961                        *
04815 00962P 0389 6F   04        A        CLR   RCBLNH,X INPUT BYTES HIGH=0
04820 00963P 038B E6   07        A        LDB   RCBMTB,X NUMBER OF TRANSFER BYTES.
04825 00964P 038D E7   05        A        STB   RCBLNL,X INPUT BYTES LOWER SET.
04830 00965P 038F 27   12   03A3          BEQ   SUB50      TRANSFER=0 BYTES.
04835 00966P 0391 10AE 02        A        LDY   RCBDBA,X INPUT DATA TRANSFER DESTINATION
04840 00967P 0394 BE   FC80      A        LDX   #SHARED    SOURCE
04845 00968                        *
04850 00969P 0397 A6   80        A SUB40  LDA   0,X+       READ DATA
04855 00970P 0399 A7   A0        A        STA   0,Y+       STORE DATA ON BUFFER
04860 00971P 039B 5A                      DECB
04865 00972P 039C 27   05   03A3          BEQ   SUB50
04870 00973P 039E 8C   FD00      A        CMPX  #SHARED+128
04875 00974P 03A1 26   F4   0397          BNE   SUB40
04880 00975P 03A3 4F               SUB50  CLRA             NORMAL
04885 00976P 03A4 BD   0610      P SUB60  JSR   REDYRQ
04890 00977P 03A7 BD   0601      P        JSR   HRESET
04895 00978P 03AA 39               SUB70  RTS
04905 00980                        ***********************************************************
04910 00981                        *
04915 00982            03AB      P INPUT  EQU   *          CONSOLE LINE INPUT
04920 00983P 03AB EC   06        A        LDD   RCBBMH,X INPUT BUFFER LENGTH SET
```

```
04925 00984P 03AD 1027 00B5 0466        LBEQ   SBTR30
04930 00985                        *
04935 00986P 03B1 FD   FFE6   A         STD    SINMAX
04940 00987P 03B4 34   10     A         PSHS   X         INDEX SAVE
04945 00988P 03B6 BD   05EA   P         JSR    BYHL      BUSY CHECK AND HALT SET
04950 00989P 03B9 108E FC80   A         LDY    #SHARED
04955 00990P 03BD 86   04     A INZ10   LDA    #GET      GET COMMAND SET
04960 00991P 03BF A7   22     A         STA    SMCMD,Y   C=4 GET COMMAND
04965 00992P 03C1 5F                    CLRB             N=0 SET
04970 00993P 03C2 EE   04     A         LDU    RCBLNH,X LENGTH GET
04975 00994                        *    OUTPUT DATA IS NOTHING ?
04980 00995P 03C4 27   18   03DE        BEQ    INZ40
04985 00996P 03C6 AE   02     A         LDX    RCBDBA,X BUFFER ADDRESS
04990 00997P 03C8 31   24     A INZ20   LEAY   4,Y       STRING ADDRESS
04995 00998P 03CA 5F                    CLRB             COUNTER CLEAR(N)
05000 00999P 03CB A6   80     A INZ25   LDA    0,X+      X POST INCREMENT
05005 01000P 03CD A7   A0     A INZ30   STA    0,Y+
05010 01001P 03CF 5C                    INCB             BYTE COUNTER UP
05015 01002P 03D0 33   5F     A         LEAU   -1,U
05020 01003P 03D2 1183 0000   A         CMPU   #0
05025 01004P 03D6 27   06   03DE        BEQ    INZ40
05030 01005P 03D8 108C FD00   A         CMPY   #SHARED+128
05035 01006P 03DC 26   ED   03CB        BNE    INZ25
05040 01007P 03DE F7   FC83   A INZ40   STB    SHARED+3
05045 01008P 03E1 4F                    CLRA
05050 01009P 03E2 1183 0000   A         CMPU   #0
05055 01010P 03E6 27   02   03EA        BEQ    INZ45
05060 01011P 03E8 86   80     A         LDA    #$80
05065 01012P 03EA BA   FC81   A INZ45   ORA    SHARED+1
05070 01013P 03ED B7   FC81   A INZ50   STA    SHARED+1 S' SET
05075 01014                        *
05080 01015P 03F0 BD   0601   P         JSR    HRESET    HALT RESET ,COMMAND OUT
05085 01016P 03F3 BD   05E5   P         JSR    BYHLWD    BUSY CHECK AND HALT SET
05090 01017P 03F6 BD   0619   P         JSR    READST
05095 01018P 03F9 26   30   042B        BNE    INP160    ERROR DETECTED
05100 01019P 03FB 1183 0000   A         CMPU   #0        END ?
05105 01020P 03FF 27   0A   040B        BEQ    INP40     YES END,GO TO INPUT
05110 01021                        *
05115 01022P 0401 108E FC80   A         LDY    #SHARED   CONTINUE
05120 01023P 0405 86   64     A         LDA    #100      C=100 SET
05125 01024P 0407 A7   22     A INZ60   STA    SMCMD,Y
05130 01025P 0409 20   BD   03C8        BRA    INZ20
05135 01026                        ***********
05140 01027P 040B 35   10     A INP40   PULS   X         ACCX COME BACK
05145 01028P 040D 34   10     A INP50   PSHS   X         ACCX SAVE
05150 01029P 040F AE   02     A         LDX    RCBDBA,X BUFFER TOP ADDRESS
05155 01030                        *  DATA IN **
05160 01031              0411   P INP60   EQU    *
05165 01032P 0411 108E FC80   A         LDY    #SHARED
05170 01033P 0415 BD   0619   P         JSR    READST    STATUS READ.
05175 01034P 0418 26   11   042B        BNE    INP160    ERROR
05180 01035                        *
05185 01036P 041A A6   23     A INP100  LDA    3,Y       K : RETURN CODE SET
05190 01037P 041C A7   80     A         STA    0,X+
05195 01038P 041E CE   0001   A         LDU    #1        READ DATA COUNT=1 INITIAL
05200 01039P 0421 8D   29   044C INP150  BSR    SBTRAN    SHARED BUFFER TRANSFER
05205 01040P 0423 26   06   042B        BNE    INP160    ERROR
05210 01041P 0425 B6   FC81   A         LDA    SHARED+1 CHECK S'
05215 01042P 0428 2B   0C   0436        BMI    INP200    MSB SET CONTINUE
05220 01043P 042A 4F                    CLRA
05225 01044P 042B BD   0610   P INP160  JSR    REDYRQ
05230 01045P 042E BD   0601   P         JSR    HRESET    HALT RESET.
05235 01046P 0431 35   10     A         PULS   X         ACCX COME BACK
05240 01047P 0433 EF   04     A         STU    RCBLNH,X READ BYTE LENGTH SET
05245 01048P 0435 39                    RTS              RETURN
05250 01049                        *
05255 01050P 0436 86   64     A INP200  LDA    #100
05260 01051P 0438 B7   FC82   A         STA    SHARED+2 CONT COMMAND SET
05265 01052P 043B BD   0601   P         JSR    HRESET    HALT RESET
05270 01053P 043E BD   05E5   P         JSR    BYHLWD    BUSY CHECK AND HALT SET
05275 01054P 0441 108E FC80   A         LDY    #SHARED
05280 01055P 0445 BD   0619   P         JSR    READST    LOAD STATUS.
05285 01056P 0448 26   E1   042B        BNE    INP160
05290 01057P 044A 20   D5   0421        BRA    INP150
05295 01058                        *
05300 01059              044C   P SBTRAN  EQU    *         SHARED MEMORY TO BUFFER
05305 01060P 044C F6   FC84   A         LDB    SHARED+4 READ BYTES LENGTH (N)
05310 01061P 044F 27   13   0464        BEQ    SBTR20
05315 01062P 0451 108E FC85   A         LDY    #SHARED+5 DATA TOP ADDRESS
05320 01063                        *
05325 01064P 0455 A6   A0     A SBTR10  LDA    0,Y+      READ DATA
05330 01065P 0457 33   41     A         LEAU   1,U       INCREMENT U
05335 01066P 0459 11B3 FFE6   A         CMPU   SINMAX    BUFFER OVER ERROR ?
05340 01067P 045D 22   07   0466        BHI    SBTR30    YES
05345 01068P 045F A7   80     A         STA    0,X+      STORE DATA
05350 01069P 0461 5A                    DECB
05355 01070P 0462 26   F1   0455        BNE    SBTR10
05360 01071P 0464 4F            SBTR20  CLRA             NORMAL.
05365 01072P 0465 39                    RTS              RETURN.
05370 01073                        *
05375 01074P 0466 86   4F     A SBTR30  LDA    #79       BUFFER OVER ERROR
05380 01075P 0468 39                    RTS
05390 01077                        *
05395 01078              0469   P INPUTC  EQU    *         INPUT CONTINUE
05400 01079P 0469 EC   06     A         LDD    RCBBMH,X BUFFER MAX LENGTH
05405 01080P 046B 27   F9   0466        BEQ    SBTR30
05410 01081P 046D FD   FFE6   A         STD    SINMAX
05415 01082                        *
05420 01083P 0470 BD   05EA   P         JSR    BYHL      BUSY CHECK AND HALT SET
05425 01084P 0473 108E FC80   A         LDY    #SHARED
05430 01085P 0477 86   05     A INZ90   LDA    #GETC     GETC COMMAND
05435 01086P 0479 A7   22     A         STA    2,Y
05440 01087P 047B BD   0601   P         JSR    HRESET    HALT RESET.
05445 01088P 047E BD   05E5   P         JSR    BYHLWD    BUSY CHECK,HALT SET.
05450 01089P 0481 20   8A   040D        BRA    INP50
05455 01090                        *
05465 01092                        ******************************************

05470 01093                        *
05475 01094              0483   P OUTPUT  EQU
05480 01095P 0483 4F                    CLRA
05485 01096P 0484 EE   04     A         LDU
05490 01097P 0486 1183 0000   A         CMPU
05495 01098P 048A 27   51   04DD        BEQ
05500 01099P 048C BD   05EA   P         JSR
05505 01100P 048F 108E FC80   A         LDY
05510 01101P 0493 86   03     A OUTZ10  LDA
05515 01102P 0495 A7   22     A         STA
05520 01103P 0497 AE   02     A         LDX
05525 01104P 0499 31   24     A OUT05   LEAY
05530 01105P 049B 5F                    CLRB
05535 01106P 049C A6   80     A OUT10   LDA
05540 01107P 049E A7   A0     A OUTZ20  STA
05545 01108P 04A0 5C                    INCB
05550 01109P 04A1 33   5F     A         LEAU
05555 01110P 04A3 1183 0000   A         CMPU
05560 01111P 04A7 27   06   04AF        BEQ
05565 01112P 04A9 108C FD00   A         CMPY
05570 01113P 04AD 26   ED   049C        BNE
05575 01114                        *
05580 01115P 04AF F7   FC83   A OUT20   STB
05585 01116P 04B2 86   00     A         LDA
05590 01117P 04B4 1183 0000   A         CMPU
05595 01118P 04B8 27   02   04BC        BEQ
05600 01119P 04BA 86   80     A         LDA
05605 01120P 04BC BA   FC81   A OUT25   ORA
05610 01121P 04BF B7   FC81   A OUTZ30  STA
05615 01122P 04C2 BD   0601   P         JSR
05620 01123P 04C5 BD   05E5   P         JSR
05625 01124P 04C8 BD   0619   P         JSR
05630 01125P 04CB 26   10   04DD        BNE
05635 01126P 04CD 1183 0000   A         CMPU
05640 01127P 04D1 27   0A   04DD        BEQ
05645 01128P 04D3 108E FC80   A         LDY
05650 01129P 04D7 86   64     A         LDA
05655 01130P 04D9 A7   22     A OUTZ40  STA
05660 01131P 04DB 20   BC   0499        BRA
05665 01132                        *
05670 01133P 04DD BD   0610   P OUT40   JSR
05675 01134P 04E0 7E   0601   P         JMP
05685 01136                        ***********
05690 01137                        *
05695 01138              04E3   P KEYIN   EQU
05700 01139P 04E3 BD   05EA   P         JSR
05705 01140P 04E6 108E FC80   A         LDY
05710 01141P 04EA 86   29     A         LDA
05715 01142P 04EC A7   22     A KEYZ10  STA
05720 01143P 04EE 86   00     A         LDA
05725 01144P 04F0 A7   23     A KEYZ20  STA
05730 01145P 04F2 BD   0601   P         JSR
05735 01146P 04F5 BD   05E5   P         JSR
05740 01147P 04F8 CE   0002   A         LDU
05745 01148P 04FB EF   04     A         STU
05750 01149P 04FD EC   23     A         LDD
05755 01150P 04FF AE   02     A         LDX
05760 01151P 0501 ED   84     A         STD
05765 01152P 0503 BD   0619   P         JSR
05770 01153P 0506 BD   0610   P         JSR
05775 01154P 0509 7E   0601   P         JMP
05785 01156              050C   P HDCOPY  EQU
05790 01157P 050C BD   0635   P         JSR
05795 01158P 050F C6   01     A         LDB
05800 01159P 0511 10AE 02     A         LDY
05805 01160                        *
05810 01161              0514   P HDLOOP  EQU
05815 01162P 0514 34   30     A         PSHS
05820 01163P 0516 E7   A9 00C8 A         STB
05825 01164P 051A C5   01     A         BITB
05830 01165P 051C 27   17   0535        BEQ
05835 01166P 051E 5C                    INCB
05840 01167P 051F 86   64     A YISETO  LDA
05845 01168P 0521 A7   A9 00CC A         STA
05850 01169P 0525 6F   A9 00CB A         CLR
05855 01170P 0529 86   C7     A         LDA
05860 01171P 052B A7   A9 00D0 A         STA
05865 01172P 052F 6F   A9 00CF A         CLR
05870 01173P 0533 20   12   0547        BRA
05875 01174                        *
05880 01175P 0535 6F   A9 00CB A YISETE  CLR
05885 01176P 0539 6F   A9 00CC A         CLR
05890 01177P 053D 6F   A9 00CF A         CLR
05895 01178P 0541 86   63     A         LDA
05900 01179P 0543 A7   A9 00D0 A         STA
05905 01180                        *
05910 01181P 0547 86   04     A XISET   LDA
05915 01182P 0549 3D                    MUL
05920 01183                        *
05925 01184P 054A 83   0008   A         SUBD
05930 01185P 054D ED   A9 00C9 A         STD
05935 01186                        *
05940 01187P 0551 C3   0007   A         ADDD
05945 01188P 0554 ED   A9 00CD A         STD
05950 01189                        * BLOCK ADDR
05955 01190                        *
05960 01191                        * DATA INPUT
05965 01192P 0558 33   A9 0000 A         LEAU
05970 01193P 055C 86   C8     A         LDA
05975 01194P 055E BD   06C7   P         JSR
05980 01195P 0561 A6   04     A         LDA
05985 01196P 0563 27   0B   0570        BEQ
05990 01197                        * BLACK INPU
05995 01198P 0565 34   10     A         PSHS
06000 01199P 0567 30   A9 0000 A         LEAX
06005 01200P 056B BD   0675   P         JSR
06010 01201P 056E 35   10     A         PULS
```

```
 *          CONSOLE OUTPUT
 RCBLNH,X LENGTH
 #0         DATA NOTHING
 OUT40
 BYHL       BUSY CHECK AND HALT SET
 #SHARED
 #PUT
 SMCMD,Y    C=3 PUT COMMAND
 RCBDBA,X BUFFER ADDRESS
 4,Y        STRING ADDRESS
            COUNTER CLEAR (N)
 0,X+
 0,Y+

 -1,U
 #0
 OUT20
 #SHARED+128
 OUT10

 SHARED+3 BYTES NUMBER
 #0
 #0
 OUT25
 #$80
 SHARED+1
 SHARED+1 S' SET
 HRESET     HALT RESET
 BYHLWD     BUSY CHECK AND HALT SET
 READST     LOAD STATUS.
 OUT40      ERROR
 #0
 OUT40      END
 #SHARED
 #100
 SMCMD,Y    C=100 SET
 OUT05

 REDYRQ     SUB READY REQUEST.
 HRESET     HALT RESET.
*******************************

 *
 BYHL       BUSY CHECK AND HALT RESET
 #SHARED
 #INKEY     INKEY COMMAND
 SMCMD,Y    C=41
 #0
 SMDAT,Y    WAITLESS
 HRESET     HALT RESET
 BYHLWD     BUSY CHECK AND HALT SET
 #2
 RCBLNH,X LENGTH SET
 SMDAT,Y
 RCBDBA,X
 0,X
 READST     LOAD STATUS.
 REDYRQ     SUB READY REQUEST.
 HRESET     HALT RESET,DOUBLE RETURN.
 *          CRT HARD COPY
 PCRLF      DUMMY OUTPUT
 #1         BLOCKNO=1 INITIAL
 RCBDBA,X GET BUFFER ADDRESS.

 *
 Y,X        BUFFER AND RCB ADDRESS SAVE
 BLKNO,Y    BLOCK NO. SAVE
 #$01       EVEN ?
 YISETE     YES EVEN

 #100       YUI=100
 PUIY+1,Y
 PUIY,Y     UPPER BYTE =0
 #199       YLI=199
 PLIY+1,Y
 PLIY,Y     UPPER BYTE =0
 XISET

 PUIY,Y     UPPER BYTE=0
 PUIY+1,Y YUI=0
 PLIY,Y     UPPER BYTE=0
 #99
 PLIY+1,Y YLI=99

 #4
            I*4

 #8         I*4-8
 PUIX,Y

 #7         PLIX
 PLIX,Y
ESS HAS BEEN COMPUTED !

 BLACK,Y    BLACK,GRAY CONTINUAS
 #200       200 BYTE CLEAR
 CPBUFC
 RCBCDT,X SOME BLACK COLOR ?
 GRAYCK     NON BLACK COLOR
T
 X          RCB ADDRESS SAVE.
 BLACK,Y
 DINPUT
 X          X:COME BACK.
```

```
01202                           *
01203P 0570 A6   05        A GRAYCK LDA   RCBCDT+1,X SOME GRAY COLOR
01204P 0572 27   07   057B          BEQ   DOUT       NON GRAY COLOR
01205                           * GRAY INPUT
01206P 0574 30   A9 0064 A          LEAX  GRAY,Y
01207P 0578 BD   0675   P           JSR   DINPUT
01208                           *
01209                           * DATA OUTPUT TO PRINTER
01210P 057B A6   A9 00C8 A DOUT     LDA   BLKNO,Y
01211P 057F 85   01        A        BITA  #$01
01212P 0581 27   03   0586          BEQ   DATAOT
01213P 0583 BD   061F   P           JSR   LFINCH     ODD,LF PITCH SET,AND BLANK OUT
01214P 0586 30   A9 0064 A DATAOT   LEAX  BLACK+100,Y BLACK=IX
01215P 058A 31   A9 00C8 A          LEAY  GRAY+100,Y GRAY=IY
01216P 058E 86   1B        A        LDA   #ESC       BI-PRINT MODE SET
01217P 0590 BD   0654   P           JSR   LPPUT
01218P 0593 86   4C        A        LDA   #'L
01219P 0595 BD   0654   P           JSR   LPPUT
01220P 0598 86   90        A        LDA   #$90       LOWER
01221P 059A BD   0654   P           JSR   LPPUT      $0190=400 DOTS
01222P 059D 86   01        A        LDA   #$01       UPPER
01223P 059F BD   0654   P           JSR   LPPUT
01224P 05A2 A6   82        A DATA01 LDA   0,-X       BLACK DATA READ
01225P 05A4 BD   0654   P           JSR   LPPUT
01226P 05A7 A6   84        A        LDA   0,X
01227P 05A9 AA   A2        A        ORA   0,-Y
01228P 05AB BD   0654   P           JSR   LPPUT      BLACK,GRAY
01229                           *
01230P 05AE A6   84        A        LDA   0,X        AGAIN SAME DATA
01231P 05B0 BD   0654   P           JSR   LPPUT
01232P 05B3 A6   84        A        LDA   0,X        POINTER INCREMENT (BLACK)
01233P 05B5 AA   A4        A        ORA   0,Y        POINTER INCREMENT (GRAY)
01234P 05B7 BD   0654   P           JSR   LPPUT
01235P 05BA AC   62        A        CMPX  2,S        100 BYTES END ?
01236P 05BC 26   E4   05A2          BNE   DATA01     NO,CONTINUE
01237                           *
01238P 05BE 35   30        A        PULS  Y,X        Y,X COME BACK
01239P 05C0 E6   A9 00C8 A          LDB   BLKNO,Y    EVEN BLOCK ?
01240P 05C4 C5   01        A        BITB  #01
01241P 05C6 26   02   05CA          BNE   *+4        NO,ODD
01242P 05C8 8D   6B   0635          BSR   PCRLF      LINEFEED OUTPUT
01243                           *
01244                           * NEXT BLOCK
01245P 05CA 5C                      INCB
01246P 05CB C1   A1        A        CMPB  #161       ALL BLOCK END ?
01247P 05CD 1026 FF43 0514          LBNE  HDLOOP     NO,CONTINUE
01248                           *
01249P 05D1 8D   06   05D9          BSR   LFP1.6     1/6 INCH LF PITCH SET.
01250P 05D3 86   0C        A        LDA   #FF        FORM FEED
01251P 05D5 8D   7D   0654          BSR   LPPUT
01252P 05D7 4F                      CLRA
01253P 05D8 39                      RTS
01254            05D9   P LFP1.6 EQU   *          ESC+A+12
01255P 05D9 86   1B        A        LDA   #ESC       $1B
01256P 05DB 8D   67   0644          BSR   LPPUTT
01257P 05DD 86   41        A        LDA   #'A        A
01258P 05DF 8D   63   0644          BSR   LPPUTT
01259P 05E1 86   0C        A        LDA   #12        12 DOTS PITCH SET.
01260P 05E3 20   5F   0644          BRA   LPPUTT
01261                           *
01262            05E5   P BYHLWD EQU   *          TIME DILAY.
01263P 05E5 86   18        A        LDA   #24        ABOUT 100 MICRO SEC.
01264P 05E7 4A              DLYLOP  DECA             2 CYCLE.
01265P 05E8 26   FD   05E7          BNE   DLYLOP     3 CYCLE.
01266            05EA   P BYHL   EQU   *          BUSY CHECK AND HALT SET
01267P 05EA 96   05        A        LDA   SBSIN      READY ?
01268P 05EC 2B   FC   05EA          BMI   BYHL       NO CONTINUE
01269                           *
01270P 05EE      1A50      A        FDB   MSKFIQ     FIRQ,IRQ IS MASKED UNTIL HALT RESET.
01271                           *
01272P 05F0 86   80        A HLSET  LDA   #$80       HALT SET
01273P 05F2 97   05        A        STA   SBSOUT     IRQREQ PRESERVED
01274                           *
01275P 05F4 34   04        A        PSHS  B
01276P 05F6 C6   08        A        LDB   #8
01277P 05F8 5A              HLTOVR  DECB
01278P 05F9 27   04   05FF          BEQ   HLTRTS
01279P 05FB 96   05        A        LDA   SBSIN
01280P 05FD 2A   F9   05F8          BPL   HLTOVR
01281P 05FF 35   84        A HLTRTS PULS  B,PC
01282                           *
01283                           *
01284            0601   P HRESET EQU   *          HALT RESET
01285P 0601 34   02        A        PSHS  A          ACCA SAVE
01286P 0603 86   00        A        LDA   #$00       IRQREQ PRESERVE
01287P 0605 97   05        A        STA   SBSOUT     HALT RESET
01288                           *
01289P 0607 1C   AF        A        ANDCC #$AF       MASK RESET.
01290                           *
01291P 0609 35   02        A        PULS  A
01292P 060B 8D   02   060F HRE10    BSR   *+4        TIME DILAY
01293P 060D 8D   00   060F          BSR   *+2
01294P 060F 39                      RTS              RETURN
01295                           *
01296            0610   P REDYRQ EQU   *
01297P 0610 F6   FC80      A        LDB   SHARED     SUB READY REQUEST.
01298P 0613 CA   80        A        ORB   #$80       MSB=1
01299P 0615 F7   FC80      A        STB   SHARED
01300P 0618 39                      RTS
01301                           *
01302            0619   P READST EQU   *
01303P 0619 B6   FC80      A        LDA   SHARED     STATUS READ.
01304P 061C 84   7F        A        ANDA  #$7F       STATUS GET.
01305P 061E 39                      RTS
01306                           *
01307            061F   P LFINCH EQU   *          LF PITCH SET
01308P 061F 86   1B        A        LDA   #ESC
```

```
06550 01309P 0621 8D   31   0654          BSR   LPPUT     LINEFEED PITCH
06555 01310P 0623 86   41      A          LDA   #'A
06560 01311P 0625 8D   2D   0654          BSR   LPPUT
06565 01312P 0627 86   08      A          LDA   #8        ESC+A+8
06570 01313P 0629 8D   29   0654          BSR   LPPUT     8/72 INCH PITCH
06575 01314P 062B C6   07      A          LDB   #7        7 SPACE
06580 01315P 062D 86   20      A LFIN10   LDA   #$20      SPACE CODE
06585 01316P 062F 8D   23   0654          BSR   LPPUT
06590 01317P 0631 5A                      DECB
06595 01318P 0632 26   F9   062D          BNE   LFIN10
06600 01319P 0634 39             LFRTS    RTS
06605 01320                      *
06610 01321                      *
06615 01322      0635          P PCRLF    EQU   *         CR LF
06620 01323P 0635 86   0A      A          LDA   #LF
06625 01324P 0637 8D   24   065D          BSR   LPPUTX
06630 01325P 0639 7D   0313    A          TST   ABORTF
06635 01326P 063C 27   F6   0634          BEQ   LFRTS     RETURN
06640 01327      063E          P ABTRTN   EQU   *
06645 01328P 063E BD   05D9    P          JSR   LFP1.6
06650 01329P 0641 7E   0000    A          JMP   ABTCHK    ABORTED.
06655 01330                      *
06660 01331      0644          P LPPUTT   EQU   *
06665 01332P 0644 34   04      A          PSHS  B
06670 01333P 0646 8E   7FFF    A          LDX   #$7FFF
06675 01334P 0649 30   1F      A LPLOOP   LEAX  -1,X
06680 01335P 064B 27   26   0673          BEQ   LPRTS
06685 01336P 064D D6   02      A          LDB   MTLPIN
06690 01337P 064F 54                      LSRB
06695 01338P 0650 25   F7   0649          BCS   LPLOOP
06700 01339P 0652 20   15   0669          BRA   LPPUT2
06705 01340                      *
06710 01341      0654          P LPPUT    EQU   *
06715 01342P 0654 34   04      A          PSHS  B
06720 01343P 0656 D6   02      A LPPUT0   LDB   MTLPIN
06725 01344P 0658 54                      LSRB
06730 01345P 0659 25   FB   0656          BCS   LPPUT0
06735 01346P 065B 20   0C   0669          BRA   LPPUT2
06740 01347                      *
06745 01348      065D          P LPPUTX   EQU   *         LP ONE BYTE OUT.
06750 01349P 065D 34   04      A          PSHS  B         B SAVE
06755 01350P 065F 7D   0313    A LPPUT1   TST   ABORTF
06760 01351P 0662 26   0F   0673          BNE   LPRTS
06765 01352P 0664 D6   02      A          LDB   MTLPIN
06770 01353P 0666 54                      LSRB            BUSY ?
06775 01354P 0667 25   F6   065F          BCS   LPPUT1
06780 01355      0669          P LPPUT2   EQU   *
06785 01356P 0669 97   01      A          STA   LPDTOT    DATA PUT
06790 01357P 066B C6   00      A          LDB   #0        STROBE PULSE
06795 01358P 066D D7   00      A          STB   MTLPOT
06800 01359P 066F CA   40      A          ORB   #$40
06805 01360P 0671 D7   00      A          STB   MTLPOT
06810 01361P 0673 35   84      A LPRTS    PULS  B,PC      B COME BACK.
06815 01362                      *
06820 01363      0675          P DINPUT   EQU   *
06825 01364P 0675 34   32      A          PSHS  A,Y,X     IX=COPY DATA(BLACK OR GRAY)
                                                          BUFFER ADDRESS.
06830 01365                      *              IY=HARD COPY BUFFER TOP ADDRESS.
06835 01366                      *              ACCA=COLOR DATA.
06840 01367P 0677 BD   05EA    P          JSR   BYHL      HALT SET
06845 01368P 067A 8E   FC80    A          LDX   #SHARED   COMMAND SET
06850 01369P 067D 86   1B      A DIN10    LDA   #GETB1    GET BLOCK 1
06855 01370P 067F A7   02      A          STA   SMCMD,X
06860 01371P 0681 EC   A9 00C9 A DIN20    LDD   PUIX,Y    PUIX STORE
06865 01372P 0685 ED   03      A          STD   3,X
06870 01373P 0687 EC   A9 00CB A DIN30    LDD   PUIY,Y    PUIY STORE
06875 01374P 068B ED   05      A          STD   5,X
06880 01375P 068D EC   A9 00CD A DIN40    LDD   PLIX,Y    PLIX STORE
06885 01376P 0691 ED   07      A          STD   7,X
06890 01377P 0693 EC   A9 00CF A DIN50    LDD   PLIY,Y    PLIY STORE
06895 01378P 0697 ED   09      A          STD   9,X
06900 01379P 0699 35   02      A          PULS  A         COLOR DATA COME BACK.
06905 01380P 069B 6F   0B      A          CLR   11,X      NUMBER OF COLOR CLEAR.
06910 01381P 069D 5F                      CLRB            COLOR CODE INITIAL=BLACK=0
06915 01382P 069E 33   0C      A          LEAU  12,X      COLOR DATA STORE TOP ADDRESS.
06920 01383                      *
06925 01384P 06A0 44             LCOLOR   LSRA            COLOR IS ON ?
06930 01385P 06A1 24   04   06A7          BCC   NCOLOR    NO OFF,NEXT COLOR
06935 01386P 06A3 6C   0B      A          INC   11,X      NUMBER OF COLOR COUNT UP
06940 01387P 06A5 E7   C0      A          STB   0,U+      COLOR CODE STORE.
06945 01388                      *              + : FOR NEXT STORE.
06950 01389P 06A7 5C             NCOLOR   INCB            COLOR CODE UP.
06955 01390P 06A8 C1   08      A          CMPB  #8        ALL BITS CHECKED ?
06960 01391P 06AA 26   F4   06A0          BNE   LCOLOR    NO,CHECK,NEXT COLOR.
06965 01392                      *
06970 01393P 06AC BD   0601    P DINP20   JSR   HRESET    HALT RESET
06975 01394P 06AF BD   05E5    P          JSR   BYHLWD    IF COMMAND END,HALT SET
06980 01395P 06B2 30   04      A          LEAX  4,X       DATA READ TOP ADDRESS.
06985 01396P 06B4 35   20      A          PULS  Y         BLACK OR GRAY : X->Y
                                                          TRANSFER.
06990 01397P 06B6 A6   80      A DINP30   LDA   0,X+      DATA READ FROM SHARED.
06995 01398P 06B8 A7   A0      A          STA   0,Y+      STORE DATA TO BLACK OR GRAY.
07000 01399P 06BA BC   FCE8    A          CMPX  #SHARED+104 100 BYTES END ?
07005 01400P 06BD 26   F7   06B6          BNE   DINP30
07010 01401P 06BF BD   0610    P          JSR   REDYRQ    SUB READY REQUEST.
07015 01402P 06C2 BD   0601    P          JSR   HRESET    HALT RESET AND DOUBLE RETURN.
07020 01403P 06C5 35   A0      A          PULS  Y,PC      Y:HARD COPY BUFFER TOP
                                                          ADDRESS.
07025 01404                      *
07030 01405      06C7          P CPBUFC   EQU   *         U : ADDRESS,A : LENGTH.
07035 01406P 06C7 6F   C0      A CPBUFL   CLR   ,U+       0 CLEAR
07040 01407P 06C9 4A                      DECA
07045 01408P 06CA 26   FB   06C7          BNE   CPBUFL    ALL END ?
07050 01409P 06CC 39                      RTS             RETURN
07055 01410                      *
07065 01412                      ********************************************
07070 01413                      *
07075 01414      06CD          P SCREEN   EQU
07080 01415P 06CD BD   0635    P          JSR
07085 01416P 06D0 10AE 02      A          LDY
07090 01417P 06D3 C6   01      A          LDB
07095 01418P 06D5 E7   A9 00C8 A          STB
07100 01419P 06D9 E7   A9 0063 A          STB
07105 01420                      *
07110 01421      06DD          P YLOOP    EQU
07115 01422P 06DD E6   A9 0063 A          LDB
07120 01423P 06E1 5A                      DECB
07125 01424P 06E2 86   08      A          LDA
07130 01425P 06E4 3D                      MUL
07135 01426                      * Y CORDINATE
07140 01427P 06E5 ED   A9 00CB A          STD
07145 01428P 06E9 C3   0007    A          ADDD
07150 01429P 06EC ED   A9 00CF A          STD
07155 01430                      * X CORDINATE
07160 01431      06F0          P XLOOP    EQU
07165 01432P 06F0 86   50      A          LDA
07170 01433P 06F2 E6   A9 00C8 A          LDB
07175 01434P 06F6 5A                      DECB
07180 01435P 06F7 3D                      MUL
07185 01436P 06F8 ED   A9 00C9 A          STD
07190 01437P 06FC C3   004F    A          ADDD
07195 01438P 06FF ED   A9 00CD A          STD
07200 01439                      * BLOCK ADDR
07205 01440P 0703 BD   0737    P          JSR
07210 01441P 0706 BD   078D    P          JSR
07215 01442P 0709 BD   07EB    P          JSR
07220 01443P 070C A6   A9 00C8 A          LDA
07225 01444P 0710 4C                      INCA
07230 01445P 0711 A7   A9 00C8 A          STA
07235 01446P 0715 81   09      A          CMPA
07240 01447P 0717 26   D7   06F0          BNE
07245 01448P 0719 86   01      A          LDA
07250 01449P 071B A7   A9 00C8 A          STA
07255 01450P 071F A6   A9 0063 A          LDA
07260 01451P 0723 4C                      INCA
07265 01452P 0724 A7   A9 0063 A          STA
07270 01453P 0728 81   1A      A          CMPA
07275 01454P 072A 26   B1   06DD          BNE
07280 01455                      * ALL END
07285 01456P 072C BD   05D9    P          JSR
07290 01457P 072F BD   0635    P          JSR
07295 01458P 0732 BD   0635    P          JSR
07300 01459P 0735 4F                      CLRA
07305 01460P 0736 39                      RTS
07310 01461                      *
07315 01462      0737          P SDINP    EQU
07320 01463P 0737 34   20      A          PSHS
07325 01464P 0739 BD   05EA    P          JSR
07330 01465P 073C 8E   FC80    A          LDX
07335 01466P 073F 86   1B      A SDIN10   LDA
07340 01467P 0741 A7   02      A          STA
07345 01468P 0743 EC   A9 00C9 A          LDD
07350 01469P 0747 ED   03      A          STD
07355 01470P 0749 EC   A9 00CB A          LDD
07360 01471P 074D ED   05      A          STD
07365 01472P 074F EC   A9 00CD A          LDD
07370 01473P 0753 ED   07      A          STD
07375 01474P 0755 EC   A9 00CF A          LDD
07380 01475P 0759 ED   09      A          STD
07385 01476                      *
07390 01477P 075B CC   0701    A          LDD
07395 01478P 075E ED   0B      A          STD
07400 01479P 0760 CC   0203    A          LDD
07405 01480P 0763 ED   0D      A          STD
07410 01481P 0765 CC   0405    A          LDD
07415 01482P 0768 ED   0F      A          STD
07420 01483P 076A CC   0607    A          LDD
07425 01484P 076D ED   88 11   A          STD
07430 01485                      *
07435 01486P 0770 BD   0601    P          JSR
07440 01487P 0773 BD   05E5    P          JSR
07445 01488P 0776 30   04      A          LEAX
07450 01489P 0778 31   A9 0000 A          LEAY
07455 01490P 077C A6   80      A SDIN30   LDA
07460 01491P 077E A7   A0      A          STA
07465 01492P 0780 BC   FCD4    A          CMPX
07470 01493P 0783 26   F7   077C          BNE
07475 01494P 0785 BD   0610    P          JSR
07480 01495P 0788 BD   0601    P          JSR
07485 01496P 078B 35   A0      A          PULS
07490 01497                      *
07495 01498      078D          P MODIFY   EQU
07500 01499P 078D 34   20      A          PSHS
07505 01500P 078F 33   A4      A          LEAU
07510 01501P 0791 30   A9 0000 A          LEAX
07515 01502P 0795 31   A9 0064 A          LEAY
07520 01503P 0799 86   0A      A          LDA
07525 01504P 079B 34   02      A DTLOOP   PSHS
07530 01505P 079D 5F                      CLRB
07535 01506P 079E A6   85      A DTLOPX   LDA
07540 01507P 07A0 8D   20   07C2          BSR
07545 01508P 07A2 CB   0A      A          ADDB
07550 01509P 07A4 C1   50      A          CMPB
07555 01510P 07A6 26   F6   079E          BNE
07560 01511P 07A8 34   10      A          PSHS
07565 01512P 07AA C6   08      A          LDB
07570 01513P 07AC 30   C9 005A A          LEAX
07575 01514P 07B0 A6   80      A DTL010   LDA
07580 01515P 07B2 A7   A0      A          STA
07585 01516P 07B4 5A                      DECB
07590 01517P 07B5 26   F9   07B0          BNE
07595 01518P 07B7 35   10      A          PULS
07600 01519P 07B9 30   01      A          LEAX
07605 01520P 07BB 35   02      A          PULS
```

```
*   SCREEN IMAGE COPY.
PCRLF     DUMMY OUTPUT
RCBDBA,X GET COPY BUFFER TOP ADDRESS.
#1
BLKNO,Y  B=1-8 SET
BLACK+99,Y A=1-25 SET

*
BLACK+99,Y GET A
          (A-1)
#8        8(A-1)

SET
PUIY,Y    PUIY SET.
#7
PLIY,Y    PLIY SET
SET
*
#80
BLKNO,Y
          80(B-1)

PUIX,Y
#79
PLIX,Y
ESS HAS BEEN COMPUTED
SDINP     BLOCK DATA INPUT.
MODIFY    DATA MODIFICATION.
PROUT     PRINTER OUTPUT.
BLKNO,Y

BLKNO,Y
#9
XLOOP     X : CONTINUE
#1        X:CONTINUE.
BLKNO,Y   X:RESET
BLACK+99,Y GET Y CORDINATE

BLACK+99,Y
#26       ALL END
YLOOP     Y : CONTINUE

LFP1.6
PCRLF     2-LINEFEED
PCRLF

          NORMAL RETURN


*         DATA INPUT
Y
BYHL      HALT SET
#SHARED   COMMAND SET
#GETB1    GET BLOCK 1
SMCMD,X
PUIX,Y    PUIX STORE
3,X
PUIY,Y    PUIY STORE
5,X
PLIX,Y    PLIX STORE
7,X
PLIY,Y    PLIY STORE
9,X

#$0701    7 COLORS ,BLUE
11,X
#$0203    RED,PURPLE
13,X
#$0405    GREEN,LIGHT BLUE
15,X
#$0607    YELLOW,WHITE.
17,X

HRESET    HALT RESET
BYHLWD    IF COMMAND END,HALT
4,X
BLACK,Y
0,X+      DATA READ.
0,Y+      STORE DATA
#SHARED+84 80 BYTE END ?
SDIN30
REDYRQ    SUB READY REQUEST.
HRESET    HALT RESET.
Y,PC

*         DATA MODIFICATION
Y
0,Y       Y->U
BLACK,Y   DATA BUFFER ADDRESS.
GRAY,Y    RESULT OF MODIFICATI
#10       COUNT=10
A         COUNTER SAVE.

B,X
SHIFT
#10
#80
DTLOPX
X         X SAVE
#8
BLACK+90,U RESULT BUFFER.
,X+
0,Y+

DTLO10
X         X COME BACK
1,X       X INCREMENT
A         COUNTER COME BACK.
```

```
01521P 07BD 4A                    DECA
01522P 07BE 26    DB   079B       BNE    DTLOOP     CONTINUE
01523P 07C0 35    A0        A     PULS   Y,PC       RETURN
01524                             *
01525             07C2      P SHIFT  EQU    *
01526P 07C2 48                    ASLA
01527P 07C3 69    C9 005A A       ROL    BLACK+90,U 1TH BYTE
01528P 07C7 48                    ASLA
01529P 07C8 69    C9 005B A       ROL    BLACK+91,U 2TH BYTE
01530P 07CC 48                    ASLA
01531P 07CD 69    C9 005C A       ROL    BLACK+92,U 3TH BYTE
01532P 07D1 48                    ASLA
01533P 07D2 69    C9 005D A       ROL    BLACK+93,U 4TH BYTE
01534P 07D6 48                    ASLA
01535P 07D7 69    C9 005E A       ROL    BLACK+94,U 5TH BYTE
01536P 07DB 48                    ASLA
01537P 07DC 69    C9 005F A       ROL    BLACK+95,U 6 TH BYTE
01538P 07E0 48                    ASLA
01539P 07E1 69    C9 0060 A       ROL    BLACK+96,U 7 TH BYTE
01540P 07E5 48                    ASLA
01541P 07E6 69    C9 0061 A       ROL    BLACK+97,U 8 TH BYTE
01542P 07EA 39                    RTS
01543                             *
01544             07EB      P PROUT  EQU    *
01545P 07EB A6    A9 00CB A       LDA    BLKNO,Y    FIRST BLOCK OF LINE ?
01546P 07EF 4A                    DECA
01547P 07F0 26    0D   07FF       BNE    NOTFST     NO,NOT FIRST.
01548P 07F2 BD    061F      P     JSR    LFINCH     LF PITCH SET
01549P 07F5 C6    03        A     LDB    #3         SPACE 10 PUT.
01550P 07F7 86    20        A     LDA    #$20
01551P 07F9 BD    0654      P SPCLOP JSR    LPPUT
01552P 07FC 5A                    DECB              ALL END ?
01553P 07FD 26    FA   07F9       BNE    SPCLOP     NO,CONTINUE
01554                             *
01555             07FF      P NOTFST EQU    *
01556                             * LDA BLKNO,Y GET BLOCK NO.
01557                             * CMPA #1
01558                             * BEQ ESCOUT
01559                             * CMPA #5
01560                             * BNE BITOUT
01561P 07FF 86    1B        A ESCOUT LDA    #ESC       BI-PRINT MODE SET.
01562P 0801 BD    0654      P     JSR    LPPUT
01563P 0804 86    4C        A     LDA    #'L        ESC L
01564P 0806 BD    0654      P     JSR    LPPUT
01565P 0809 86    50        A     LDA    #$50       LOWER=80 DOTS.
01566P 080B BD    0654      P     JSR    LPPUT
01567P 080E 86    00        A     LDA    #0         UPPER=0
01568P 0810 BD    0654      P     JSR    LPPUT
01569                             *
01570P 0813 30    A9 0064 A BITOUT LEAX   GRAY,Y     LP PUT DATA TOP ADDRE
01571P 0817 C6    50        A     LDB    #80        COUNTER SET.
01572P 0819 A6    80        A NEXLPD LDA    0,X+       GET A DATA
01573P 081B BD    0654      P     JSR    LPPUT      PRINT OUT
01574P 081E 5A                    DECB              ALL DATA END ?
01575P 081F 26    F8   0819       BNE    NEXLPD     NO,NEXT LP DATA.
01576P 0821 A6    A9 00CB A       LDA    BLKNO,Y
01577P 0825 81    08        A     CMPA   #8         LAST BLOCK OF LINE?
01578P 0827 26    03   082C       BNE    NOCRLF
01579P 0829 BD    0635      P     JSR    PCRLF      DOUBLE RETURN
01580P 082C 39                NOCRLF RTS               RETURN
01582             082D      P VPARM  EQU    *          VOICE PARAMETER SET
01583P 082D 8D    25   0854       BSR    VREADY
01584P 082F EE    04        A     LDU    RCBLNH,X   GET BYTES LENGTH
01585P 0831 1183  0000      A     CMPU   #0         LENGTH=0 DQ NOTHING
01586P 0835 27    1B   0852       BEQ    VPAR50     ERORR
01587                             *
01588P 0837 10AE  02        A     LDY    RCBDBA,X   GET BUFFER ADDRESS
01589P 083A A6    A0        A VPAR10 LDA    0,Y+       READ PARAMETER DATA
01590P 083C 97    09        A     STA    VOICMD     DATA OUTPUT
01591                             *
01592P 083E 86    04        A     LDA    #$04       INT ACTIVE
01593P 0840 97    08        A     STA    VOICNT
01594                             *
01595P 0842 8D    1E   0862       BSR    VBUSY
01596                             *
01597P 0844 86    00        A     LDA    #$00       INT INACTIVE
01598P 0846 97    08        A     STA    VOICNT
01599P 0848 8D    0A   0854       BSR    VREADY
01600                             *
01601P 084A 33    5F        A     LEAU   -1,U       DECREMENT U.
01602P 084C 1183  0000      A     CMPU   #0         ALL END ?
01603P 0850 26    E8   083A       BNE    VPAR10     NO,CONTINUE
01604P 0852 4F                VPAR50 CLRA              NORMAL
01605P 0853 39                    RTS               RETURN
01606                             *
01607P 0854 8E    FFFF      A VREADY LDX    #$FFFF
01608P 0857 30    1F        A VRELOP LEAX   -1,X
01609P 0859 27    06   0861       BEQ    VRERTS     Z=1 ZERO.
01610P 085B 96    0A        A     LDA    VOICIN
01611P 085D 2A    F8   0857       BPL    VRELOP
01612P 085F C6    01        A     LDB    #1         Z(>1 NON ZERO.
01613P 0861 39                VRERTS RTS
01614                             *
01615P 0862 8E    FFFF      A VBUSY  LDX    #$FFFF
01616P 0865 30    1F        A VBLOP  LEAX   -1,X
01617P 0867 27    06   086F       BEQ    VBRTS      Z=1 ZERO.
01618P 0869 96    0A        A     LDA    VOICIN
01619P 086B 2B    F8   0865       BMI    VBLOP
01620P 086D C6    01        A     LDB    #1         Z(>1 NON ZERO
01621P 086F 39                VBRTS  RTS
01623             0870      P VDPUT  EQU    *          VOICE DATA TRANSFER
01624P 0870 86    20        A     LDA    #$20
01625P 0872 97    08        A     STA    VOICNT     RAM COUNTER CLEAR
01626                             *
01627P 0874 EE    04        A     LDU    RCBLNH,X   GET TRANSFER BYTES LENGTH
01628P 0876 1183  0000      A     CMPU   #0         LENGTH=0 DO NOTHING
01629P 087A 27    1B   0897       BEQ    VDPU20
```

```
08155 01630P 087C 86    02          A          LDA    #2          ASSUME RCB ERROR.
08160 01631P 087E 1183  0800        A          CMPU   #$0800
08165 01632P 0882 22    14    0898             BHI    VDPU30      TOO LONG
08170 01633                                    *
08175 01634P 0884 10AE  02          A          LDY    RCBDBA,X    GET DATA BUFFER ADDRESS
08180 01635P 0887 A6    A0          A VDPU10   LDA    0,Y+        READ VOICE DATA
08185 01636P 0889 97    0B          A          STA    VOICDT      STORE VOICE DATA
08190 01637                                    *
08195 01638P 088B 33    5F          A          LEAU   -1,U        DECREMENT U COUNTER
08200 01639P 088D 1183  0000        A          CMPU   #0          ALL END ?
08205 01640P 0891 26    F4    0887             BNE    VDPU10      NO,CONTINUE
08210 01641P 0893 86    20          A          LDA    #$20
08215 01642P 0895 97    08          A          STA    VOICNT
08220 01643P 0897 4F                  VDPU20   CLRA               NORMAL
08225 01644P 0898 39                  VDPU30   RTS                RETURN
08235 01646             0899        P VOICE    EQU    *           VOICE OUTPUT
08240 01647P 0899 86    01          A          LDA    #$01        VOICE SPEAKER OPEN
08245 01648P 089B 97    03          A          STA    BELL
08250 01649                                    *
08255 01650P 089D 8D    B5    0854             BSR    VREADY
08260 01651P 089F 86    21          A          LDA    #$21        *START=ACTIVE
08265 01652P 08A1 97    08          A          STA    VOICNT
08270 01653                                    *
08275 01654P 08A3 8D    BD    0862             BSR    VBUSY
08280 01655P 08A5 86    00          A          LDA    #$00        *START=INACTIVE
08285 01656P 08A7 97    08          A          STA    VOICNT
08290 01657                                    *
08295 01658P 08A9 96    0A          A VOILOP   LDA    VOICIN      READY ?
08300 01659P 08AB 2A    FC    08A9             BPL    VOILOP
08305 01660                                    *
08310 01661P 08AD 4F                           CLRA
08315 01662P 08AE 39                           RTS                NORMAL,RETURN
08320 01663                                    *
08325 01664             08AF        P VRESET   EQU    *           RESET SYNTHESIZER
08330 01665P 08AF 86    28          A          LDA    #$28        COUNTER CLEAR,DSP RESET
08335 01666P 08B1 97    08          A          STA    VOICNT
08340 01667P 08B3 86    19          A          LDA    #25
08345 01668P 08B5 4A                  RSTLP    DECA
08350 01669P 08B6 26    FD    08B5             BNE    RSTLP
08355 01670P 08B8 97    08          A          STA    VOICNT
08360 01671P 08BA 8D    03    08BF             BSR    WAIT
08365 01672P 08BC 8D    96    0854             BSR    VREADY
08370 01673P 08BE 4F                           CLRA               NORMAL RETURN
08375 01674P 08BF 39                  WAIT     RTS
08385 01676             08C0        P BIINIT   EQU    *           BIOS INITIAL SETTING
08390 01677P 08C0 86    01          A          LDA    #$01
08395 01678P 08C2 97    03          A          STA    BELL
08400 01679P 08C4 4F                           CLRA
08405 01680P 08C5 39                           RTS
08410 01681                                    *** CRT HARD COPY WORK
08415 01682             0000        A BLACK    EQU    $00         BLACK DATA BUFFER  OFFSET.
08420 01683             0064        A GRAY     EQU    $64         GRAY DATA BUFFER OFFSET.
08425 01684             00CB        A BLKNO    EQU    $CB         CRT DATA BLOCK NO.
08430 01685                                    *
08435 01686             00C9        A PUIX     EQU    $C9         PUIX ADDRESS
08440 01687             00CB        A PUIY     EQU    $CB         PUIY ADDRESS
08445 01688                                    *
08450 01689             00CD        A PLIX     EQU    $CD         PLIX ADDRESS
08455 01690             00CF        A PLIY     EQU    $CF         PLIY ADDRESS
08460 01691                                    *
08465 01692             FBFC        A BSRES    EQU    $FBFC       BASIC RESTART ADDRESS.
                                                                  DO NOT INIT.
08470 01693             FBFE        A ROMENT   EQU    $FBFE       BASIC START ADDRESS.FROM
                                                                  INITIALIZE.

08479 01695                                    OPT    L
08480 01696A FE00                              ORG    $FE00
08485 01697             FE00        A BOTENT   EQU    *           F-BASIC BOOT PROGRAM.
08490 01698A FE00 20    09    FE0B             BRA    STARTS
08495 01699                                    *
08500 01700             FC7F        A BOTSTK   EQU    $FC7F
08505 01701             0100        A IPLADR   EQU    $0100
08510 01702A FE02 7E    FE67        A RESTOR   JMP    RRESTR
08515 01703A FE05 7E    FF6F        A DWRITE   JMP    RDWRIT
08520 01704A FE08 7E    FF6F        A DREAD    JMP    RDREAD
08525 01705             FE0B        A STARTS   EQU    *
08530 01706A FE0B CC    FFFC        A          LDD    #NMIRTI     NMI VECTOR SET.
08535 01707A FE0E FD    FFFC        A          STD    PNMI
08540 01708A FE11 86    FD          A          LDA    #$FD
08545 01709A FE13 1F    8B          A          TFR    A,DP        DIRECT PAGE=FD.
08550 01710A FE15 10CE  FC7F        A          LDS    #BOTSTK     STACK SET.
08555 01711             FE19        A ROMVBT   EQU    *
08560 01712                                    *
08565 01713A FE19 D6    04          A          LDB    STOPF
08570 01714A FE1B C5    02          A          BITB   #$02        STOP KEY PRESSED ?
08575 01715A FE1D 26    05    FE24             BNE    DISKV       DISK CODE LOAD TRY.
08580 01716A FE1F BE    FBFC        A          LDX    BSRES       YES,JUMP TO WARM START ENTRY.
08585 01717A FE22 20    23    FE47             BRA    SYSJMP
08590 01718                                    *
08595 01719             FE24        A DISKV    EQU    *           TRY TO READ DISK CODE.
08600 01720A FE24 96    05          A          LDA    SBSOUT
08605 01721A FE26 44                           LSRA
08610 01722A FE27 25    18    FE41             BCS    NONDS2
08615 01723A FE29 C6    05          A          LDB    #5          RETRY COUNTS=5 SET.
08620 01724A FE2B 34    04          A LRETRY   PSHS   B
08625 01725A FE2D 8E    FE57        A          LDX    #BOTRCB     RCB ADDRESS FOR BOOT.
08630 01726A FE30 8D    D6    FE08             BSR    DREAD       FLOPPY READ ROUTINE.
08635 01727A FE32 35    04          A          PULS   B           GET RETRY COUNTS
08640 01728A FE34 8E    0100        A          LDX    #IPLADR
08645 01729A FE37 4D                           TSTA               ANY ERROR ?
08650 01730A FE38 27    0D    FE47             BEQ    SYSJMP      DISK IPL LOAD COMPLETED.
08655 01731A FE3A 81    0A          A          CMPA   #$0A        NOT READY ?
08660 01732A FE3C 27    03    FE41             BEQ    NONDS2
08665 01733                                    *
```

```
08670 01734A FE3E 5A                  NONDSK
08675 01735A FE3F 26    0B    FE4C
08680 01736A FE41 BE    FBFE        A NONDS2
08685 01737A FE44 4F
08690 01738A FE45 97    1D          A
08695 01739                           *
08700 01740A FE47 4F                  SYSJMP
08705 01741A FE48 1F    8B          A
08710 01742A FE4A 6E    84          A
08715 01743                           *
08720 01744A FE4C 34    04          A LRETZZ
08725 01745A FE4E 8E    FE5F        A
08730 01746A FE51 8D    AF    FE02
08735 01747A FE53 35    04          A
08740 01748A FE55 20    D4    FE2B
08745 01749                           *
08750 01750             FE57        A BOTRCB
08755 01751A FE57       0A          A
08760 01752A FE58       00          A
08765 01753A FE59       0100        A
08770 01754A FE5B       00          A
08775 01755A FE5C       01          A
08780 01756A FE5D       00          A
08785 01757A FE5E       00          A
08790 01758                           *
08795 01759             FE5F        A RETRCB
08800 01760A FE5F       08          A
08805 01761A FE60       0000        A
08810 01762A FE66       00          A
08820 01764                           *
08825 01765                           ******
08830 01766                           *
08835 01767                           *****
08840 01768                           *
08845 01769             FE67        A RRESTR
08850 01770A FE67 CE    FFE1        A
08855 01771A FE6A A6    07          A
08860 01772A FE6C 84    03          A
08865 01773A FE6E 6F    C6          A RRESEN
08870 01774A FE70 D6    1D          A
08875 01775A FE72 8A    80          A
08880 01776A FE74 97    1D          A
08885 01777A FE76 8D    4C    FEC4
08890 01778A FE78 96    1D          A
08895 01779A FE7A 2A    4D    FEC9
08900 01780A FE7C 5D
08905 01781A FE7D 2B    08    FE87
08910 01782A FE7F 86    02          A
08915 01783A FE81 BD    FF4D        A Z500
08920 01784A FE84 4A
08925 01785A FE85 26    FA    FE81
08930 01786A FE87 86    0A          A Z1000
08935 01787A FE89 20    1A    FEA5
08940 01788                           *
08945 01789             FE8B        A SEEK
08950 01790A FE8B 8D    7B    FF08
08955 01791A FE8D 25    3A    FEC9
08960 01792A FE8F CE    FFE1        A
08965 01793A FE92 F6    FFE8        A
08970 01794A FE95 A6    C5          A
08975 01795A FE97 97    19          A
08980 01796A FE99 A6    04          A
08985 01797A FE9B A1    C5          A
08990 01798A FE9D 27    24    FEC3
08995 01799A FE9F A7    C5          A
09000 01800A FEA1 97    1B          A
09005 01801A FEA3 86    1E          A
09010 01802                           *
09015 01803                           *
09020 01804             FEA5        A CMEXEC
09025 01805A FEA5 97    18          A
09030 01806A FEA7 86    03          A
09035 01807                           *
09040 01808A FEA9 4A                  CME02
09045 01809A FEAA 27    31    FEDD
09050 01810A FEAC 108E  C800        A CME05
09055 01811A FEB0 31    3F          A CME10
09060 01812A FEB2 27    F5    FEA9
09065 01813                           *
09070 01814A FEB4 D6    1F          A
09075 01815A FEB6 C5    40          A
09080 01816A FEB8 27    F6    FEB0
09085 01817A FEBA D6    18          A
09090 01818                           *
09095 01819A FEBC BD    FF64        A
09100 01820                           *
09105 01821A FEBF C5    18          A
09110 01822A FEC1 26    08    FECB
09115 01823A FEC3 4F                  CME11
09120 01824A FEC4 8D    02    FEC8    CME15
09125 01825A FEC6 8D    00    FEC8
09130 01826A FEC8 39
09135 01827                           *
09140 01828A FEC9 C6    80          A CME16
09145 01829A FECB 86    0A          A CME20
09150 01830A FECD C5    80          A
09155 01831A FECF 26    0E    FEDF
09160 01832A FED1 86    0C          A
09165 01833A FED3 C5    10          A
09170 01834A FED5 26    08    FEDF
09175 01835A FED7 86    0D          A
09180 01836A FED9 C5    08          A
09185 01837A FEDB 26    02    FEDF
09190 01838A FEDD 86    0F          A CME25
09195 01839A FEDF 1A    01            CME30
09200 01840A FEE1 39
09205 01841                           *
```

```
 DECB
 BNE     LRETZZ
 LDX     ROMENT
 CLRA
 STA     DDRIVE    MOTOR STOP.

 CLRA              ROM BASIC INTERFACE=ACCA=0
 TFR     A,DP      DP=0
 JMP     0,X       GO TO EACH ENTRY.

 PSHS    B
 LDX     #RETRCB
 BSR     RESTOR
 PULS    B
 BRA     LRETRY

 EQU     *
 FCB     FDREAD    RQNO
 FCB     0         STATUS
 FDB     IPLADR    DATA STORE ADDRESS.
 FCB     0         TRACK=0
 FCB     1         SECTOR=1
 FCB     0         SIDE=0
 FCB     0         UNIT=0

 EQU     *
 FCB     FDREST    RQNO
 FDB     0,0,0
 FCB     0         UNIT 0 SET.

******************************************

YPE 1 COMMAND

EQU     *         SEEK TRACK 0.
LDU     #CURTRK
LDA     RCBUNT,X
ANDA    #$03      UNIT SELECT.
CLR     A,U       TRACK ADDRESS 0 SET.
LDB     DDRIVE
ORA     #$80
STA     DDRIVE
BSR     CME15
LDA     DDRIVE
BPL     CME16
TSTB
BMI     Z1000
LDA     #2
JSR     TM1S
DECA
BNE     Z500
LDA     #$0A      RESTORE COMMAND (V=0)
BRA     CMEXEC    BRANCH COMMAND EXECUTION

EQU     *         SEEK TRACK
BSR     DRSEL     DRIVE SELECT
BCS     CME16
LDU     #CURTRK
LDB     DSTDRV
LDA     B,U
STA     DTRACK
LDA     RCBTRK,X DESTINATION TRACK LOAD.
CMPA    B,U
BEQ     CME11     NON SEEK ACTION
STA     B,U       NEW TRACK ADDRESS SET.
STA     DDATA     TRACK NO IN DATA REG
LDA     #$1E      SEEK COMMAND SET (V=1)


EQU     *         COMMAND EXECUTION
STA     DCMD      COMMAND STORE
LDA     #3

DECA              TIME OVER ?
BEQ     CME25     YES,TIME OVER.
LDY     #$C800    ABOUT 2.2 SECOND
LEAY    -1,Y
BEQ     CME02

LDB     DDRQ      READ DRQ,IRQ
BITB    #$40      IRQREQ ?
BEQ     CME10     BUSY,LOOP
LDB     DSTAT     MAKE SURE

JSR     TM20M

BITB    #$18      CRC,SEEK ERROR ?
BNE     CME20     ERROR OCCURED
CLRA
BSR     *+4
BSR     *+2
RTS

LDB     #$80
LDA     #$0A      NOT READY ?
BITB    #$80
BNE     CME30     YES
LDA     #$0C      SEEK ERROR
BITB    #$10
BNE     CME30     YES
LDA     #$0D      CRC ERROR ?
BITB    #$08
BNE     CME30     YES
LDA     #$0F      TIME OVER ERROR
SEC
RTS
```

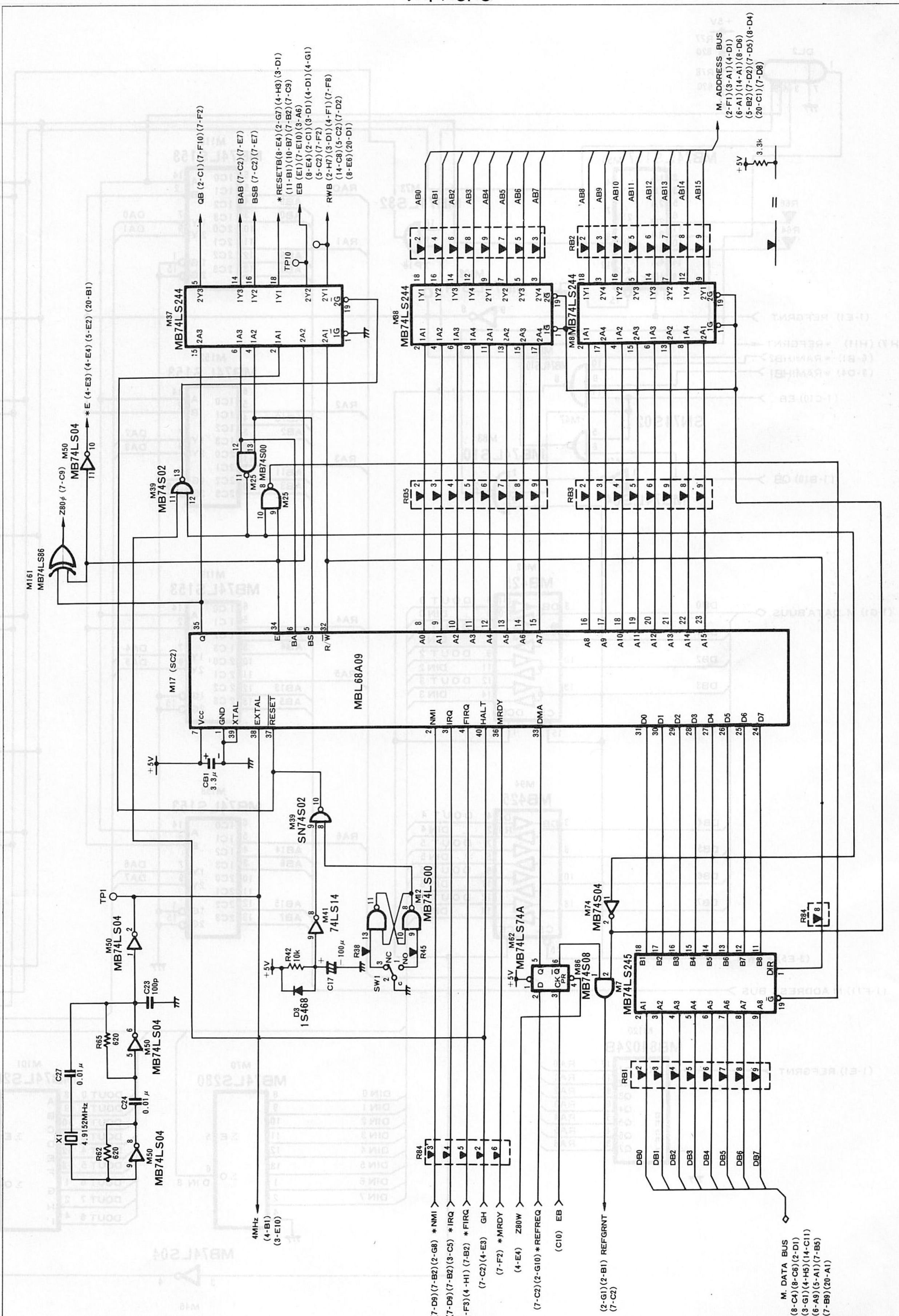
```
01842                FEE2    A FREADY EQU    *
01843A FEE2 D6   18          A        LDB    DSTAT
01844A FEE4 54                        LSRB            READY ? FDC READY CHECK.
01845A FEE5 24   04   FEEB            BCC    YESRDY
01846                          * FORCE INTERRUPT FDC NOT READY.
01847A FEE7 C6   D0          A        LDB    #$D0     FORCE INTERRUPT COMMAND.
01848A FEE9 D7   18          A        STB    DCMD
01849                          * FDC READY
01850A FEEB 96   1B          A YESRDY LDA    DDATA
01851A FEED 43                        COMA
01852A FEEE 97   1B          A        STA    DDATA
01853A FEF0 8D   D2   FEC4            BSR    CME15
01854A FEF2 91   1B          A        CMPA   DDATA
01855A FEF4 26   CD   FEC3            BNE    CME11
01856A FEF6 CE   5000        A        LDU    #$5000
01857A FEF9 33   5F          A RDYCHK LEAU   -1,U     TIME OUT ?
01858A FEFB 1183 0000        A        CMPU   #0
01859A FEFF 27   06   FF07            BEQ    TIMOUT   YES TIMEOUT Z=1 ZERO.
01860A FF01 D6   18          A        LDB    DSTAT
01861A FF03 2B   F4   FEF9            BMI    RDYCHK   Z=0 NON ZERO SET.
01862A FF05 C6   01          A        LDB    #1
01863A FF07 39                 TIMOUT RTS
01865                          ******** DRIVE SELECT
01866                          *
01867                FF08    A DRSEL  EQU    *
01868A FF08      1A50        A        FDB    MSKFIQ   FIRQ,IRQ MASK.
01869A FF0A D6   1D          A        LDB    DDRIVE   READ CURRENT DRIVE.
01870A FF0C F7   FFE5        A        STB    DMOTOR
01871A FF0F C4   03          A        ANDB   #$03     MSB=MOTOR FLAG 1=ON 0=OFF.
01872A FF11 F7   FFE0        A        STB    CRDRIV   CURRENT DRIVE NO SET.
01873A FF14 CE   FFE1        A        LDU    #CURTRK  CURRENT HEAD POSITION SAVE AREA.
01874A FF17 A6   07          A        LDA    RCBUNT,X
01875A FF19 84   03          A        ANDA   #$03     DESTINATION UNIT SELECT.
01876A FF1B B7   FFE8        A        STA    DSTDRV
01877A FF1E 7D   FFE5        A        TST    DMOTOR   WAS ON ?
01878A FF21 2B   17   FF3A            BMI    WASON    TES,IT WAS MOTOR ON.
01879                          * NO,MOTOR MUST BE ON AND WAIT 1 SEC AND CLEAR THE
                                 TRACK POSITION SAVE
01880                          * AREA.  FORCE RESTORE ALL DRIVES.
01881                          *
01882A FF23 86   80          A DVTCLR LDA    #$80
01883A FF25 34   42          A RESTLP PSHS   A,U      DRIVE NO SAVE.
01884A FF27 BD   FE6E        A        JSR    RRESEN   CALL RESTORE ROUTINE.
01885A FF2A 35   42          A        PULS   A,U      DRIVE NO PULL DOWN.
01886A FF2C 4C                        INCA
01887A FF2D 81   84          A        CMPA   #$84     ALL END ?
01888A FF2F 26   F4   FF25            BNE    RESTLP   NO,CONTINUE.
01889                          *
01890A FF31 B6   FFE8        A        LDA    DSTDRV
01891A FF34 8A   80          A        ORA    #$80
01892A FF36 97   1D          A        STA    DDRIVE   DESTINATION DRIVE SET.
01893A FF38 20   09   FF43            BRA    CHKRDY
01894                          *
01895A FF3A B1   FFE0        A WASON  CMPA   CRDRIV   DESTINATION=CURRENT ?
01896A FF3D 27   04   FF43            BEQ    CHKRDY   YES
01897A FF3F 8A   80          A        ORA    #$80
01898A FF41 97   1D          A        STA    DDRIVE   IT IS NOT SAME. DRIVE REG SET.
01899                          *
01900A FF43 8D   9D   FEE2 CHKRDY BSR    FREADY   READY CHECK.
01901A FF45 27   03   FF4A            BEQ    NOTRDY   NOT READY.
01902A FF47 1C   FE                   CLC             NORMAL
01903A FF49 39                        RTS
01904                          *
01905A FF4A 1A   01            NOTRDY SEC             NOT READY.
01906A FF4C 39                        RTS
01907                          *
01908                FF4D    A TM1S   EQU    *        1 SECOUND TIMER=985
01909A FF4D CE   03D9        A        LDU    #$03D9
01910A FF50 8D   09   FF5B TM1S5  BSR    TM1M
01911A FF52 33   5F          A        LEAU   -1,U
01912A FF54 1183 0000        A        CMPU   #0
01913A FF58 26   F6   FF50            BNE    TM1S5
01914A FF5A 39                        RTS
01915                          *
01916A FF5B 108E 0099        A TM1M   LDY    #$99     1 MS TIMER.
01917A FF5F 31   3F          A TM1M5  LEAY   -1,Y     DEC Y
01918A FF61 26   FC   FF5F            BNE    TM1M5
01919A FF63 39                        RTS
01920                          *
01921A FF64 34   04          A TM20M  PSHS   B
01922A FF66 C6   14          A        LDB    #20
01923A FF68 8D   F1   FF5B TM20LP BSR    TM1M
01924A FF6A 5A                        DECB
01925A FF6B 26   FB   FF68            BNE    TM20LP
01926A FF6D 35   84          A        PULS   B,PC
01928                          **********************************
01929                          *
01930                FF6F    A RDWRIT EQU    *        SECTOR WRITE.
01931                FF6F    A RDREAD EQU    *
01932A FF6F BD   FEBB        A        JSR    SEEK     DRIVE SELECT ,SEEK TRACK
01933A FF72 26   58   FFCC            BNE    DREAD7
01934A FF74 A6   06          A        LDA    RCBSID,X
01935A FF76 84   01          A        ANDA   #1       GET UNIT.
01936A FF78 97   1C          A        STA    DSIDE
01937A FF7A A6   05          A        LDA    RCBSCT,X
01938A FF7C 97   1A          A        STA    DSECTR   SECTOR SET
01939A FF7E A6   84          A        LDA    RQNO,X   GET REQUEST NO.
01940A FF80 81   0A          A        CMPA   #FDREAD  DREAD ?
01941A FF82 27   18   FF9C            BEQ    DREADX   YES ,BRANCH DREAD.
01942A FF84 86   A0          A        LDA    #$A0     WRITE COMMAND SET
01943A FF86 97   18          A        STA    DCMD
01944A FF88 10AE 02          A        LDY    RCBDBA,X
01945A FF8B A6   A0          A DWRIT1 LDA    0,Y+     DATA GET
01946A FF8D D6   1F          A DWRIT2 LDB    DDRQ     DRQ ?
01947A FF8F 2A   04   FF95            BPL    DWRIT5   NO
01948A FF91 97   1B          A        STA    DDATA    DATA STORE
01949A FF93 20   F6   FF8B            BRA    DWRIT1
```

```
                                   *
01951A FF95 58                     DWRIT5 ASLB             EXTRACT COMMAND END TO MASB.
01952A FF96 2A      F5    FFBD            BPL    DWRIT2
01953A FF98 BD      C1    FF5B            BSR    TM1M
01954A FF9A 20      14    FFB0            BRA    RWEND
                                   ******************************************
                                   *
                    FF9C        A  DREADX EQU    *         SECTOR READ
01959A FF9C 86      80          A         LDA    #$80      READ COMMAND SET
01960A FF9E 97      1B          A         STA    DCMD
01961A FFA0 10AE    02          A         LDY    RCBDBA,X
01962A FFA3 D6      1F          A  DREAD1 LDB    DDRQ      DRQ ?
01963A FFA5 2A      06    FFAD            BPL    DREAD5    NO
01964A FFA7 96      1B          A         LDA    DDATA     LOAD DATA
01965A FFA9 A7      A0          A         STA    0,Y+
01966A FFAB 20      F6    FFA3            BRA    DREAD1

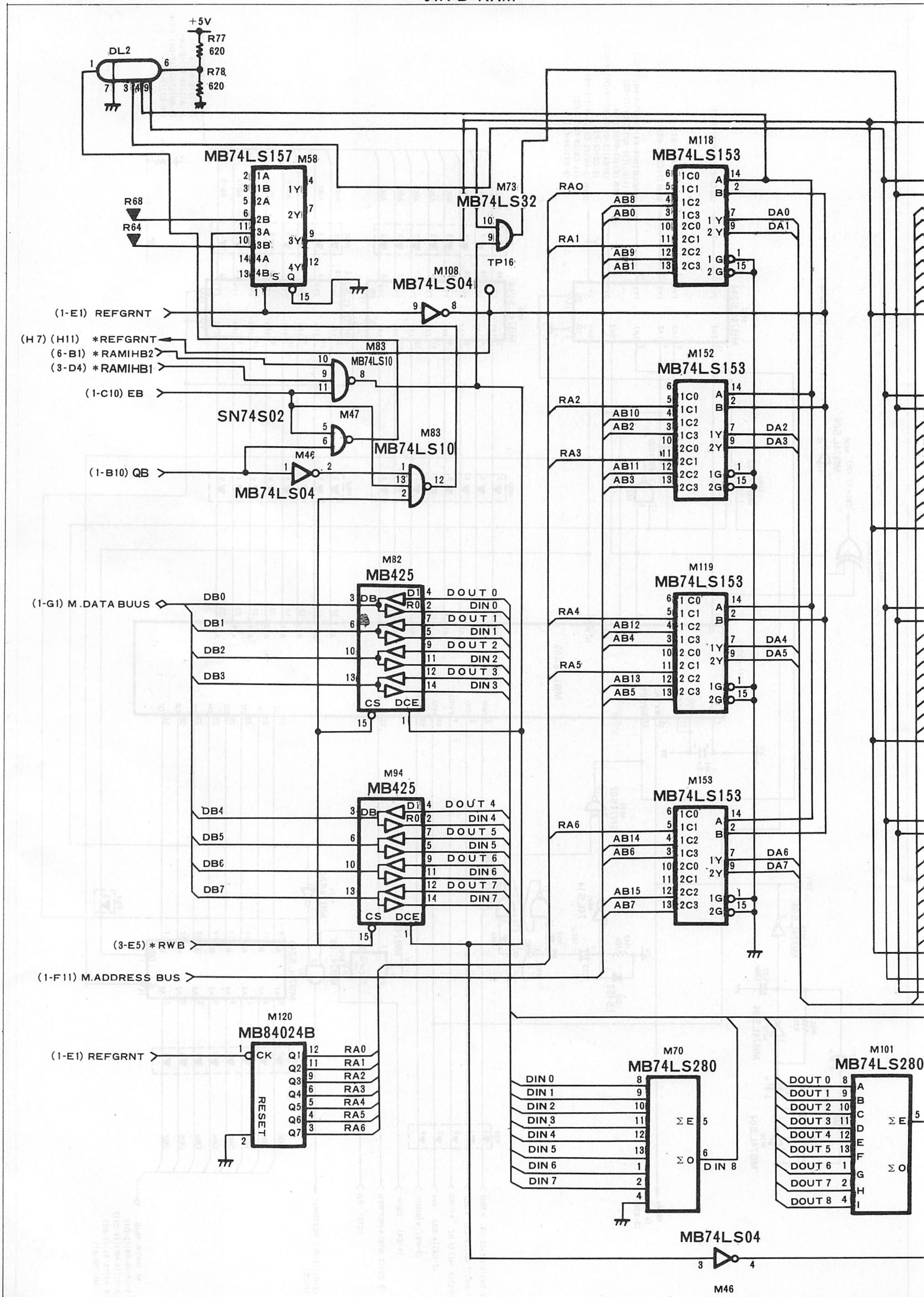
01968A FFAD 58                     DREAD5 ASLB             EXTRACT COMMAND END TO MSB.
01969A FFAE 2A      F3    FFA3            BPL    DREAD1
01970A FFB0 D6      18          A  RWEND  LDB    DSTAT     ERROR ?
01971A FFB2 26      02    FFB6            BNE    DREAD6
01972A FFB4 4F                            CLRA
01973A FFB5 39                            RTS
                                   *
01975A FFB6 86      0A          A  DREAD6 LDA    #$0A
01976A FFB8 C5      80          A         BITB   #$80      NOT READY ?
01977A FFBA 26      10    FFCC            BNE    DREAD7
01978A FFBC 4C                            INCA             $0B
01979A FFBD C5      40          A         BITB   #$40      WRITE PROTECTED ?
01980A FFBF 26      0B    FFCC            BNE    DREAD7
01981A FFC1 4C                            INCA             $0C
01982A FFC2 C5      14          A         BITB   #$14      RECORD NOT FOUND,LOST DATA ?
01983A FFC4 26      06    FFCC            BNE    DREAD7
01984A FFC6 4C                            INCA             $0D CRC ERROR ?
01985A FFC7 C5      08          A         BITB   #$08
01986A FFC9 26      01    FFCC            BNE    DREAD7
01987A FFCB 4C                            INCA             $0E DELETED DATA MARK DETECTED
01988A FFCC 1A      01             DREAD7 SEC              SET CARRY FOR DRSEL ERROR
                                                           INTERFACE.
01989A FFCE 39                            RTS

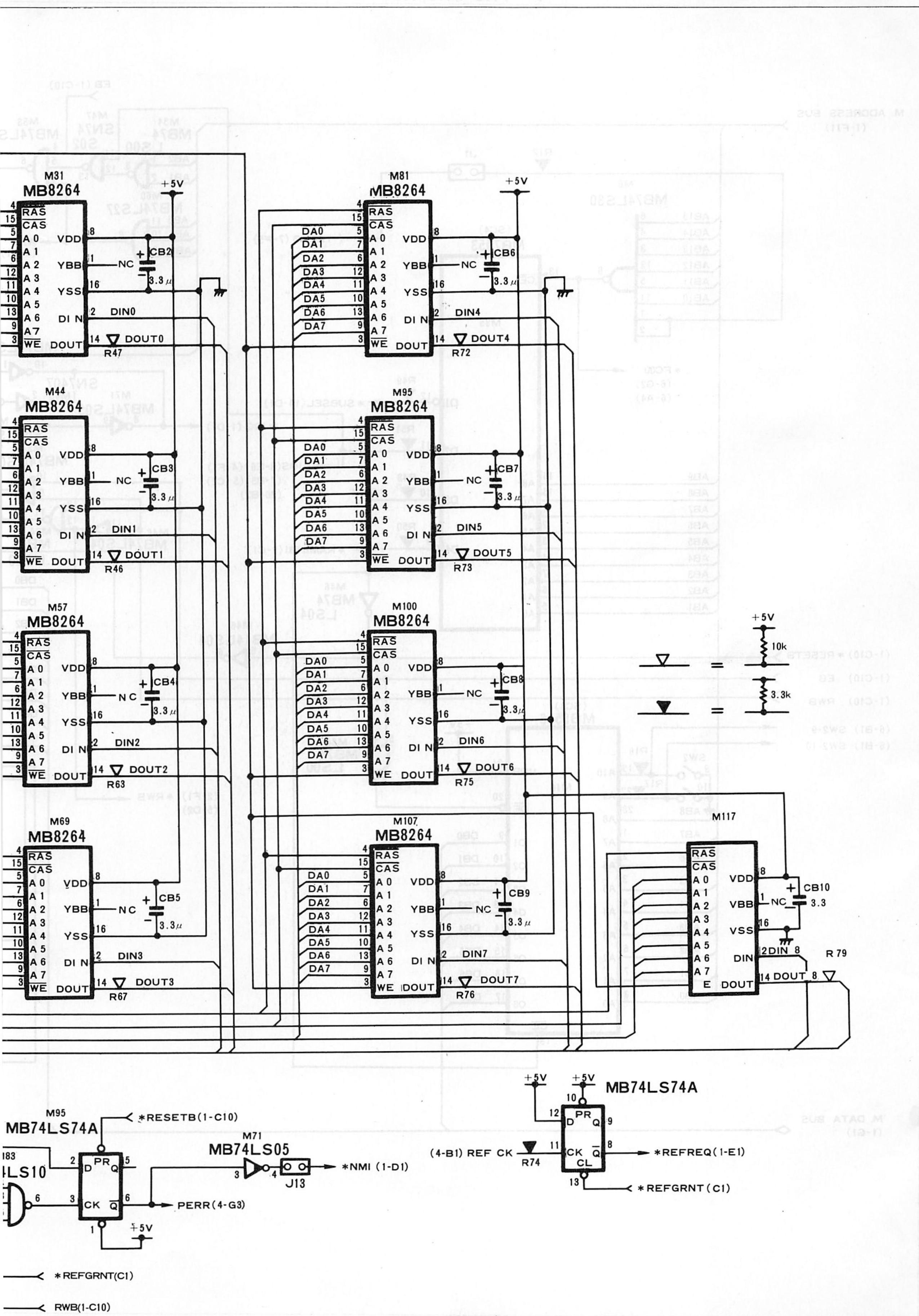
01991A FFCF 3B                     NMIRTI RTI              DO NOTHING.
                                   * BOOT ROM END ADDRESS=$FFDF
                                   *
01994A FFFE                               ORG    $FFFE
01995A FFFE         FE00        A         FDB    BOTENT    BOOT ENTRY ADDRESS.
                                   *
                                          END
TOTAL ERRORS 00000--00000
TOTAL WARNINGS 00000--00000
```

メインCPU

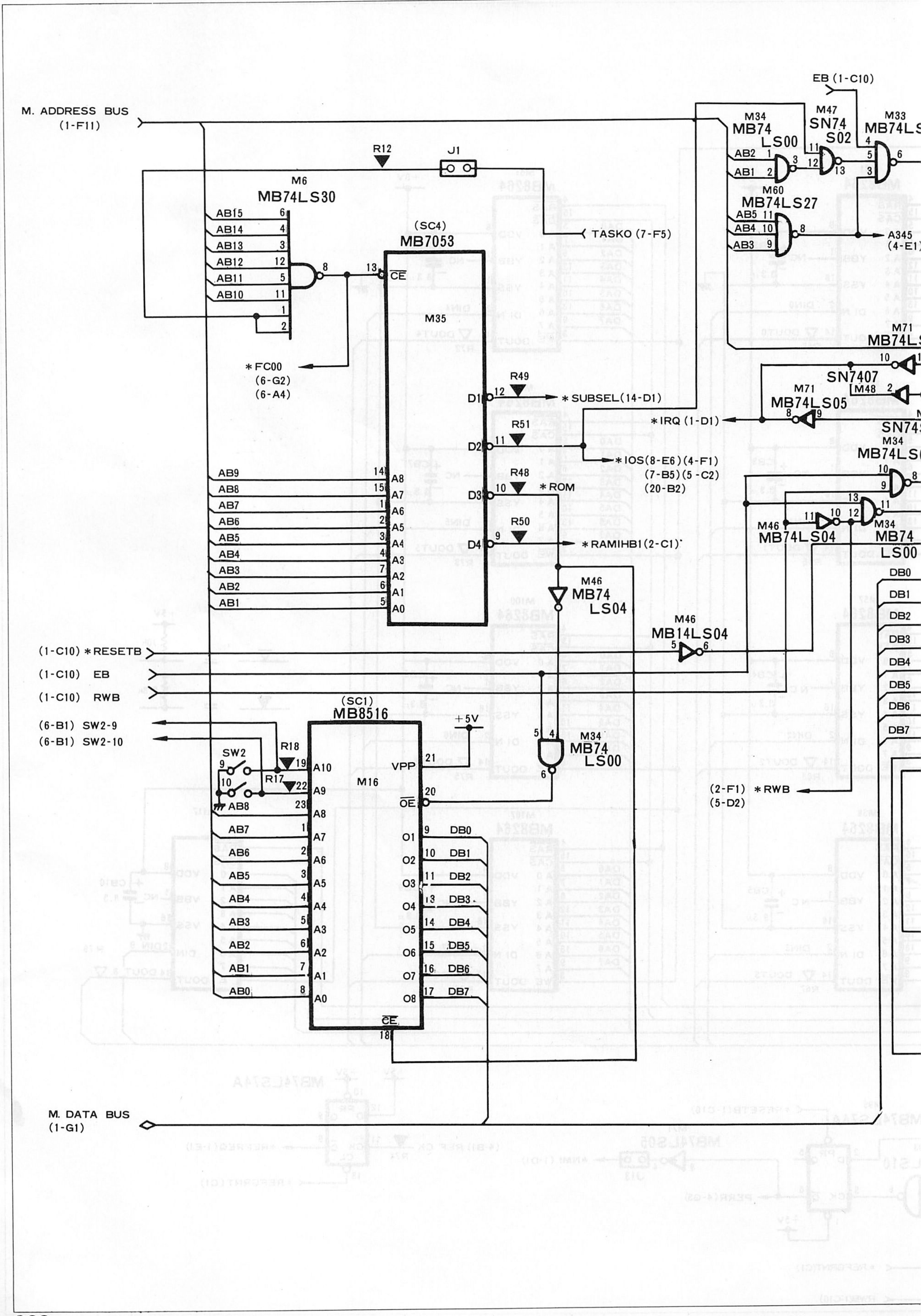

MBL68A09
M17 (SC2)
MB74LS244
M37
M88
M8
MB74LS245
M7
MB74S02
M39
SN74S02
MB74S00
M25
MB74LS00
M12
MB74LS04
M50
M74
74LS14
M41
MB74LS74A
M62
MB74S08
M86
MB74LS86
M161
M. ADDRESS BUS
M. DATA BUS
4.9152MHz
1S468

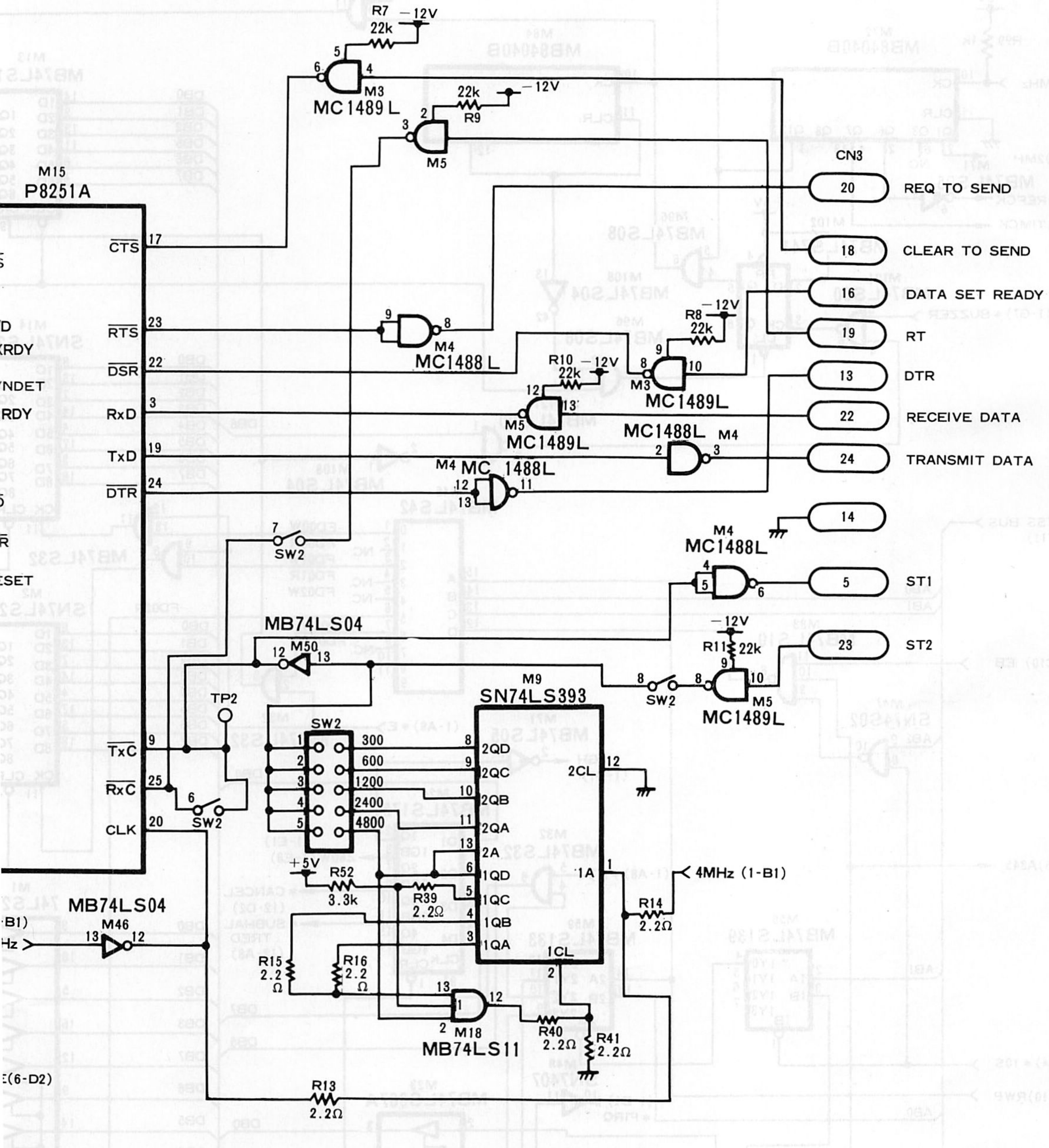
64K D-RAM


+5V
R77
620
R78
620
DL2
MB74LS157 M58
R68
R64
MB74LS32 M73
TP16
M108
MB74LS04
(1-E1) REFGRNT
(H7) (H11) *REFGRNT
(6-B1) *RAMIHB2
(3-D4) *RAMIHB1
(1-C10) EB
M83
MB74LS10
SN74S02 M47
M83
MB74LS10
(1-B10) QB
M46
MB74LS04
M82
MB425
(1-G1) M.DATA BUUS
DB0
DB1
DB2
DB3
DOUT 0
DIN 0
DOUT 1
DIN 1
DOUT 2
DIN 2
DOUT 3
DIN 3
M94
MB425
DB4
DB5
DB6
DB7
DOUT 4
DIN 4
DOUT 5
DIN 5
DOUT 6
DIN 6
DOUT 7
DIN 7
(3-E5) *RWB
(1-F11) M.ADDRESS BUS
M120
MB84024B
(1-E1) REFGRNT
RESET
RA0
RA1
RA2
RA3
RA4
RA5
RA6
M118
MB74LS153
AB8
AB0
AB9
AB1
DA0
DA1
M152
MB74LS153
AB10
AB2
AB11
AB3
DA2
DA3
M119
MB74LS153
AB12
AB4
AB13
AB5
DA4
DA5
M153
MB74LS153
AB14
AB6
AB15
AB7
DA6
DA7
M70
MB74LS280
DIN 8
M101
MB74LS280
DOUT 0
DOUT 1
DOUT 2
DOUT 3
DOUT 4
DOUT 5
DOUT 6
DOUT 7
DOUT 8
MB74LS04
M46


M31
MB8264
M44
MB8264
M57
MB8264
M69
MB8264
M81
MB8264
M95
MB8264
M100
MB8264
M107
MB8264
M117
RAS
CAS
A 0
A 1
A 2
A 3
A 4
A 5
A 6
A 7
WE
VDD
YBB
YSS
DI N
DOUT
NC
+5V
CB2
CB3
CB4
CB5
CB6
CB7
CB8
CB9
CB10
3.3 μ
DIN0
DIN1
DIN2
DIN3
DIN4
DIN5
DIN6
DIN7
DOUT0
DOUT1
DOUT2
DOUT3
DOUT4
DOUT5
DOUT6
DOUT7
R47
R46
R63
R67
R72
R73
R75
R76
R79
DA0
DA1
DA2
DA3
DA4
DA5
DA6
DA7
10k
3.3k
MB74LS74A
PR
CL
CK
*REFREQ(1-E1)
(4-B1) REF CK
R74
*REFGRNT(C1)
M95
MB74LS74A
*RESETB(1-C10)
M71
MB74LS05
J13
*NMI (1-D1)
PERR(4-G3)
*REFGRNT(C1)
RWB(1-C10)

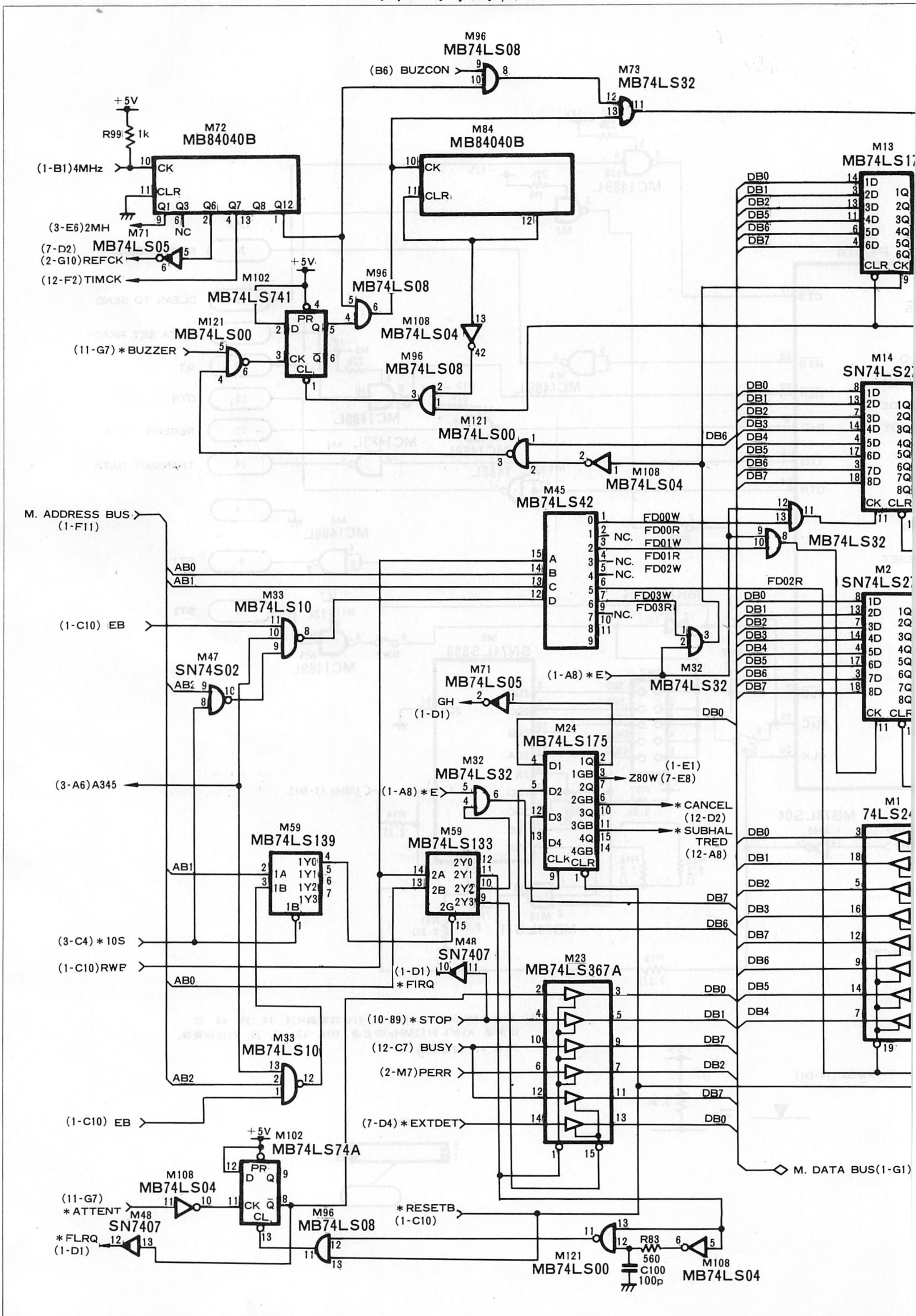
ブートROM/RS-232C

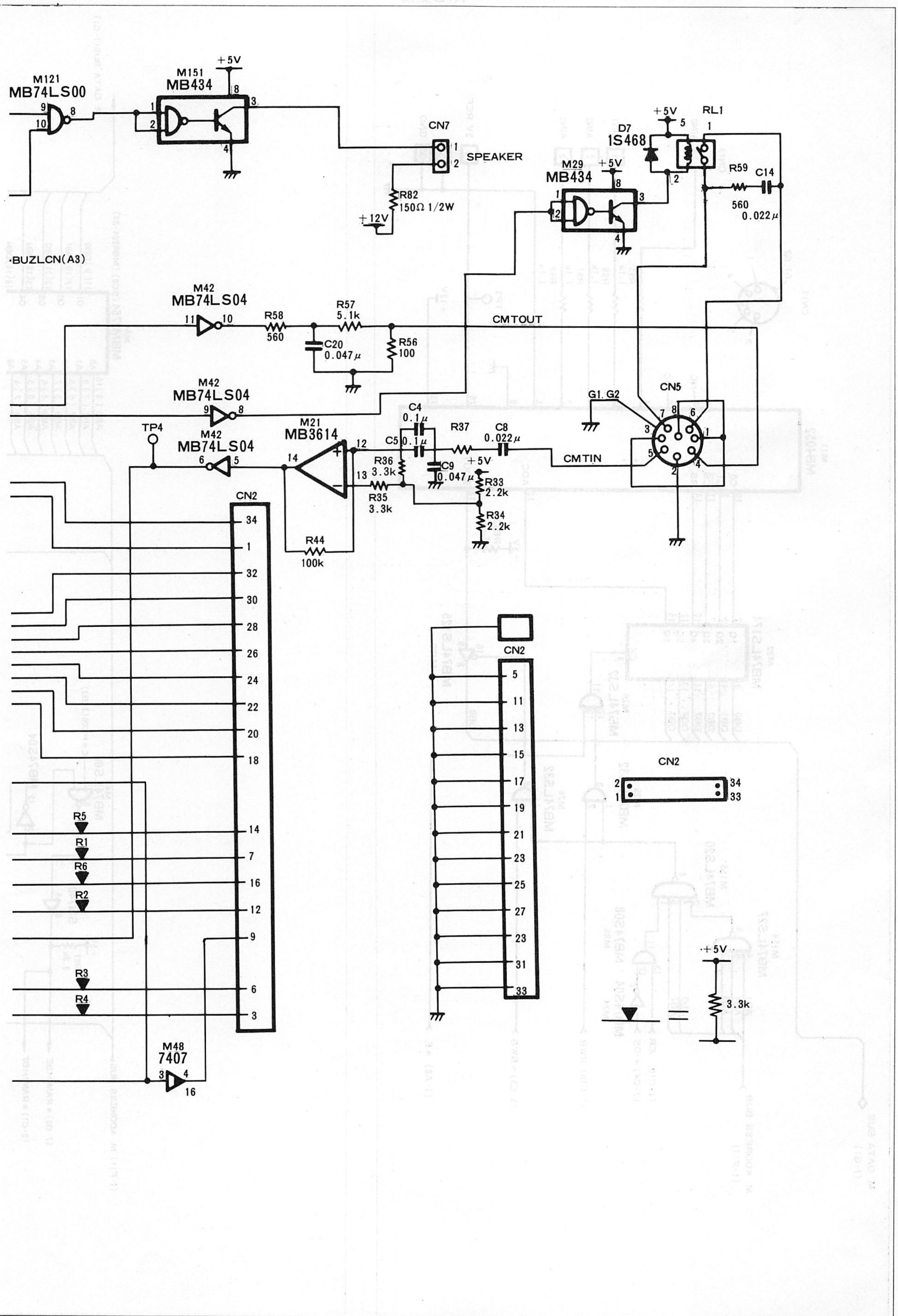




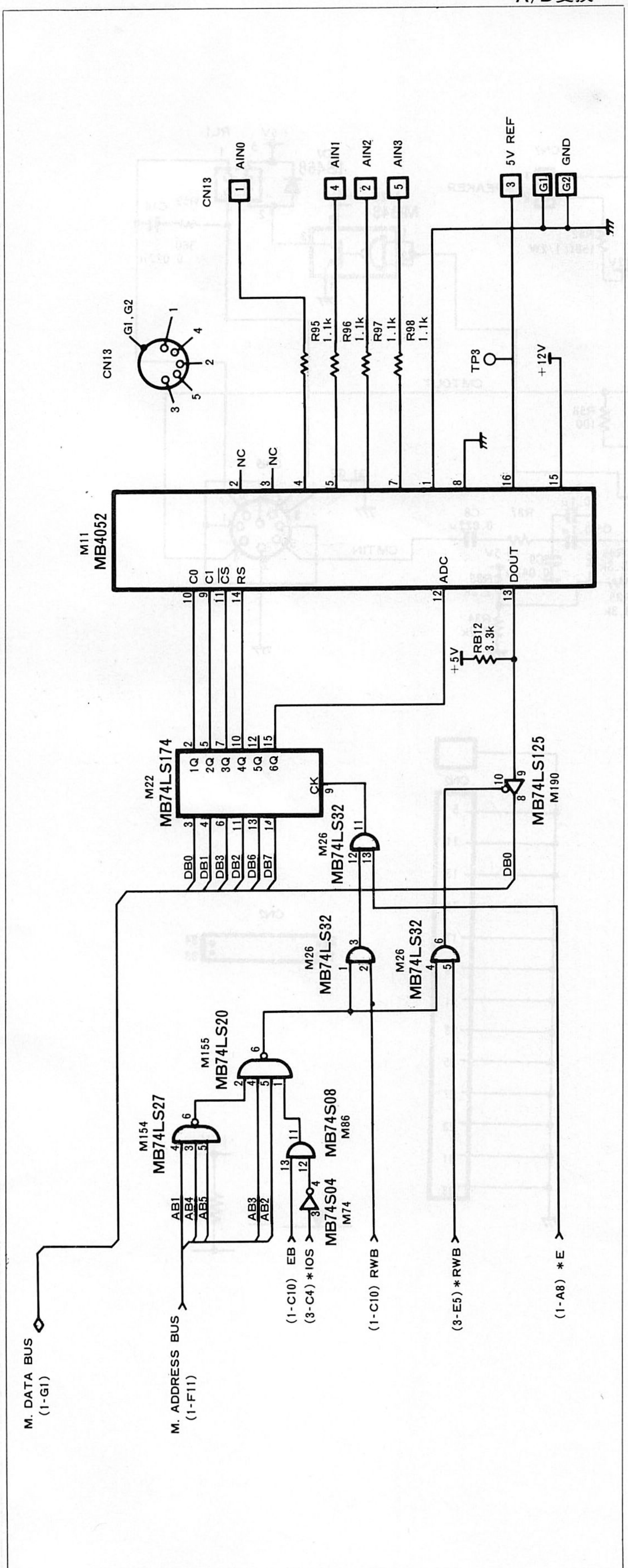
（注） X1が7MHzのときR13, 15, 41は非實装R14, 16, 39, 40, 52は実装　X1が4.9152MHzのとき，R13, 15, 16, 39, 40は非実装，R14, 41, 52は実装．

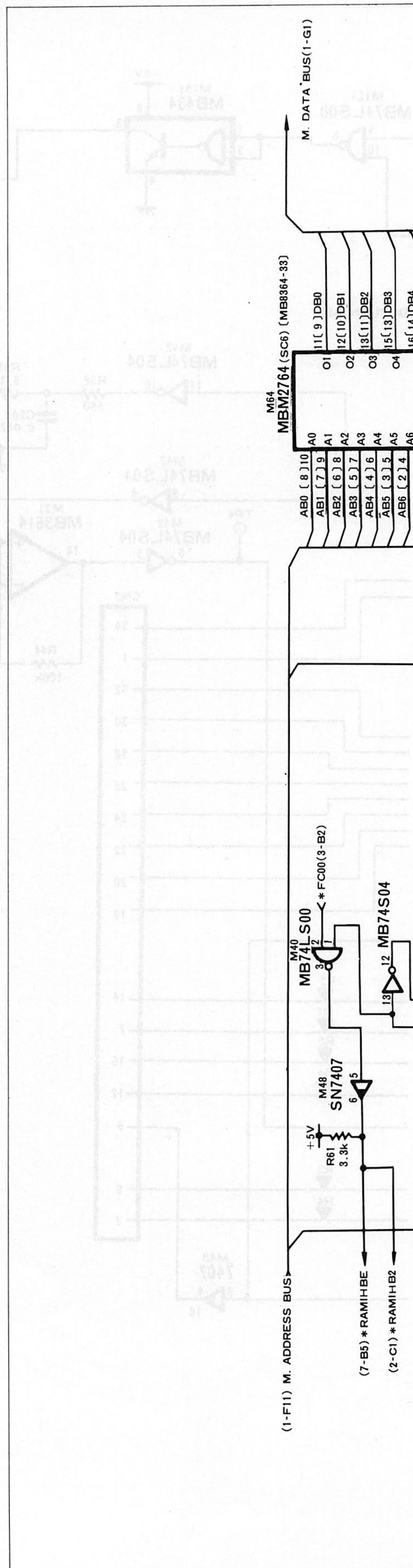
ライン・プリンタ/リニア


M96
MB74LS08
(B6) BUZCON
M73
MB74LS32
+5V
R99 1k
M72
MB84040B
M84
MB84040B
(1-B1)4MHz
CK
CLR
Q1 Q3 Q6 Q7 Q8 Q12
(3-E6)2MH
M71
NC
M13
MB74LS17
(7-D2)
(2-G10)REFCK
MB74LS05
(12-F2)TIMCK
M102
MB74LS741
M96
MB74LS08
M121
MB74LS00
(11-G7) *BUZZER
M108
MB74LS04
M96
MB74LS08
M121
MB74LS00
M108
MB74LS04
M14
SN74LS2
M45
MB74LS42
M. ADDRESS BUS
(1-F11)
AB0
AB1
FD00W
FD00R
FD01W
FD01R
FD02W
NC.
MB74LS32
FD02R
M2
SN74LS2
FD03W
FD03R
M33
MB74LS10
(1-C10) EB
M47
SN74S02
AB2
M32
MB74LS32
(1-A8) *E
M71
MB74LS05
GH
(1-D1)
M24
MB74LS175
M32
MB74LS32
Z80W (7-E8)
(1-E1)
*CANCEL
(12-D2)
*SUBHAL
TRED
(12-A8)
(3-A6)A345
M59
MB74LS139
M59
MB74LS133
M1
74LS24
(3-C4) *10S
(1-C10)RWE
M48
SN7407
(1-D1)
*FIRQ
M23
MB74LS367A
M33
MB74LS10
(10-89) *STOP
(12-C7) BUSY
(2-M7)PERR
(1-C10) EB
(7-D4) *EXTDET
M. DATA BUS(1-G1)
+5V
M102
MB74LS74A
M108
MB74LS04
(11-G7)
*ATTENT
M48
SN7407
*FLRQ
(1-D1)
M96
MB74LS08
*RESETB
(1-C10)
R83
560
C100
100p
M121
MB74LS00
M108
MB74LS04
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7

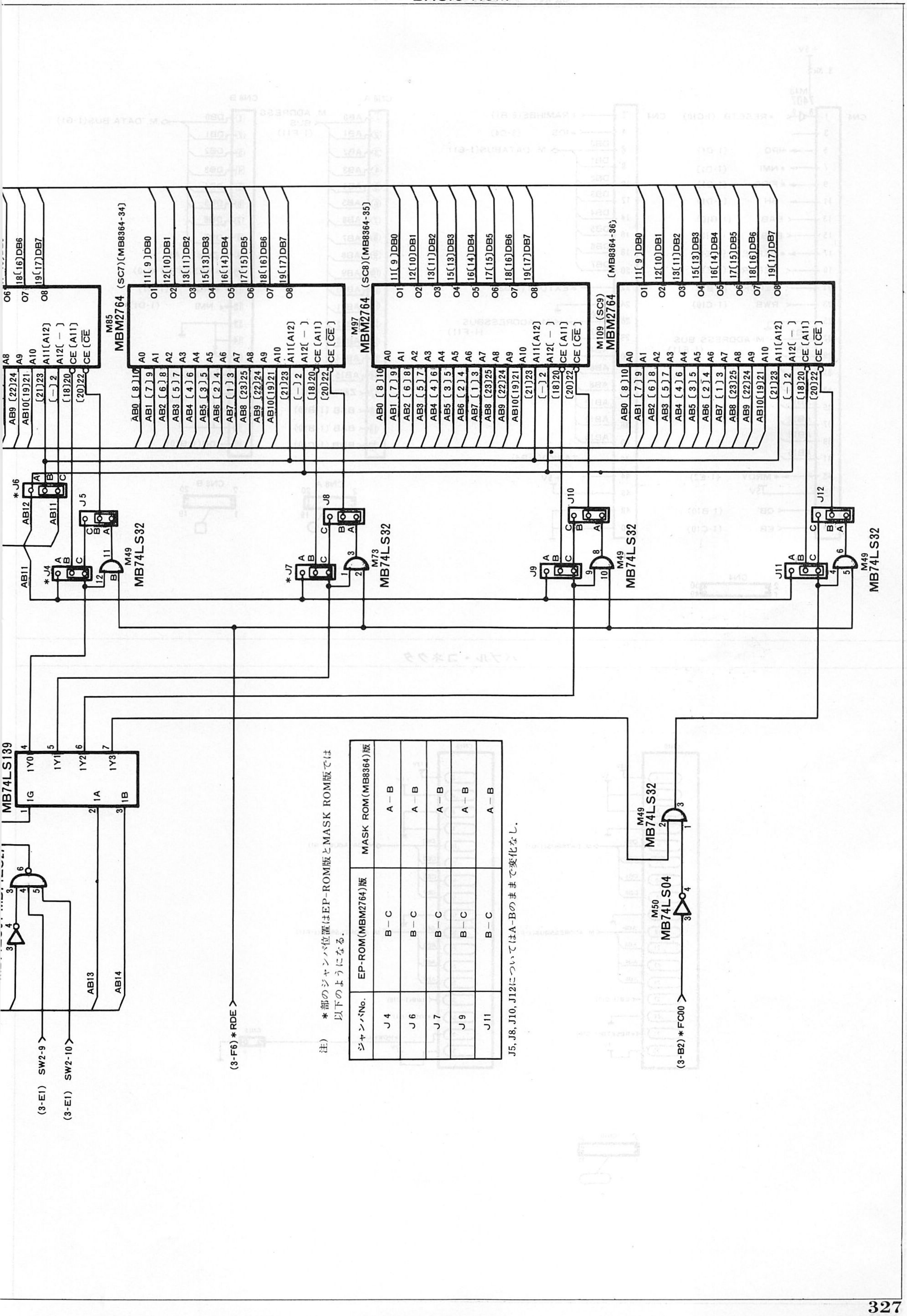

M121
MB74LS00
M151
MB434
+5V
CN7
SPEAKER
R82
150Ω 1/2W
+12V
+5V
RL1
D7
1S468
M29
MB434
R59
C14
560
0.022μ
·BUZLCN(A3)
M42
MB74LS04
R58
560
R57
5.1k
C20
0.047μ
R56
100
CMTOUT
M42
MB74LS04
G1. G2
CN5
TP4
M42
MB74LS04
M21
MB3614
C4
0.1μ
C5
0.1μ
R37
C8
0.022μ
CMTIN
R36
3.3k
C9
0.047μ
+5V
R33
2.2k
R35
3.3k
R34
2.2k
R44
100k
CN2
R5
R1
R6
R2
R3
R4
M48
7407
CN2
CN2
+5V
3.3k

A/D変換


CN13
AIN0
AIN1
AIN2
AIN3
5V REF
GND
G1,G2
R95 1.1k
R96 1.1k
R97 1.1k
R98 1.1k
TP3
+12V
M11 MB4052
NC
C0
C1
CS
RS
ADC
DOUT
+5V
RB12 3.3k
M22 MB74LS174
1Q
2Q
3Q
4Q
5Q
6Q
CK
DB0
DB1
DB3
DB2
DB6
DB7
M26 MB74LS32
MB74LS125
M190
M155 MB74LS20
M154 MB74LS27
MB74S08
M86
MB74S04
M74
AB1
AB4
AB5
AB3
AB2
M. DATA BUS (1-G1)
M. ADDRESS BUS (1-F11)
(1-C10) EB
(3-C4) *IOS
(1-C10) RWB
(3-E5) *RWB
(1-A8) *E


M. DATA BUS (1-G1)
M64
MBM2764 (SC6) (MB8364-33)
DB0
DB1
DB2
DB3
DB4
O1
O2
O3
O4
A0
A1
A2
A3
A4
A5
A6
AB0
AB1
AB2
AB3
AB4
AB5
AB6
*FC00 (3-B2)
M40 MB74LS00
MB74S04
M48 SN7407
+5V
R61 3.3k
(1-F11) M. ADDRESS BUS
(7-B5) *RAMIHBE
(2-C1) *RAMIHB2

BASIC ROM

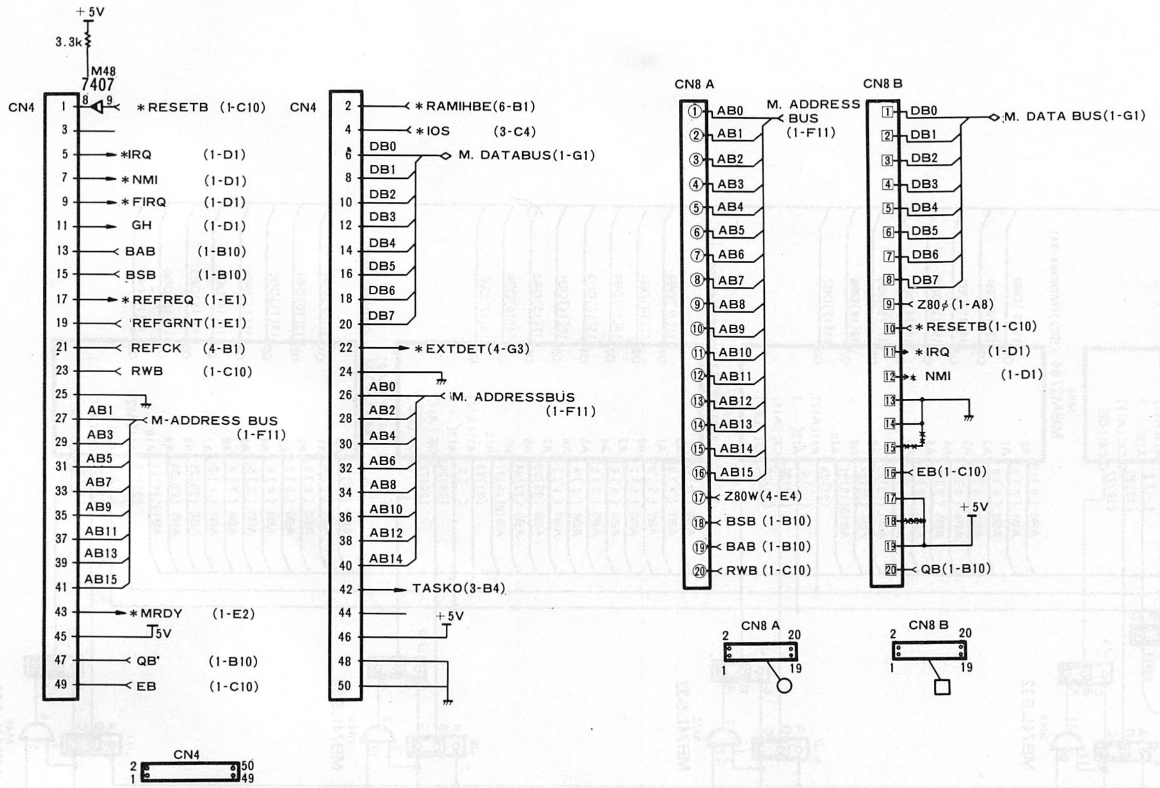


注） ＊部のジャンパ位置はEP-ROM版とMASK ROM版では以下のようになる。

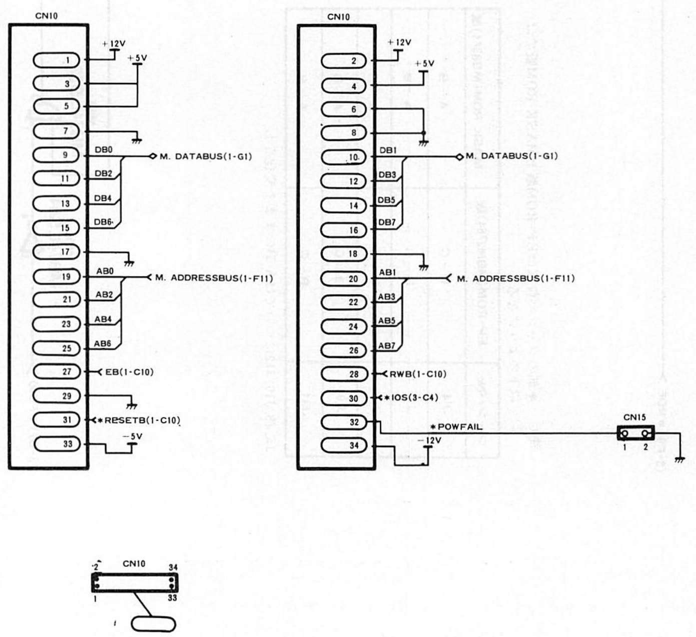
ジャンパNo.	EP-ROM(MBM2764)版	MASK ROM(MB8364)版
J 4	B − C	A − B
J 6	B − C	A − B
J 7	B − C	A − B
J 9	B − C	A − B
J 11	B − C	A − B

J5, J8, J10, J12についてはA-Bのままで変化なし。

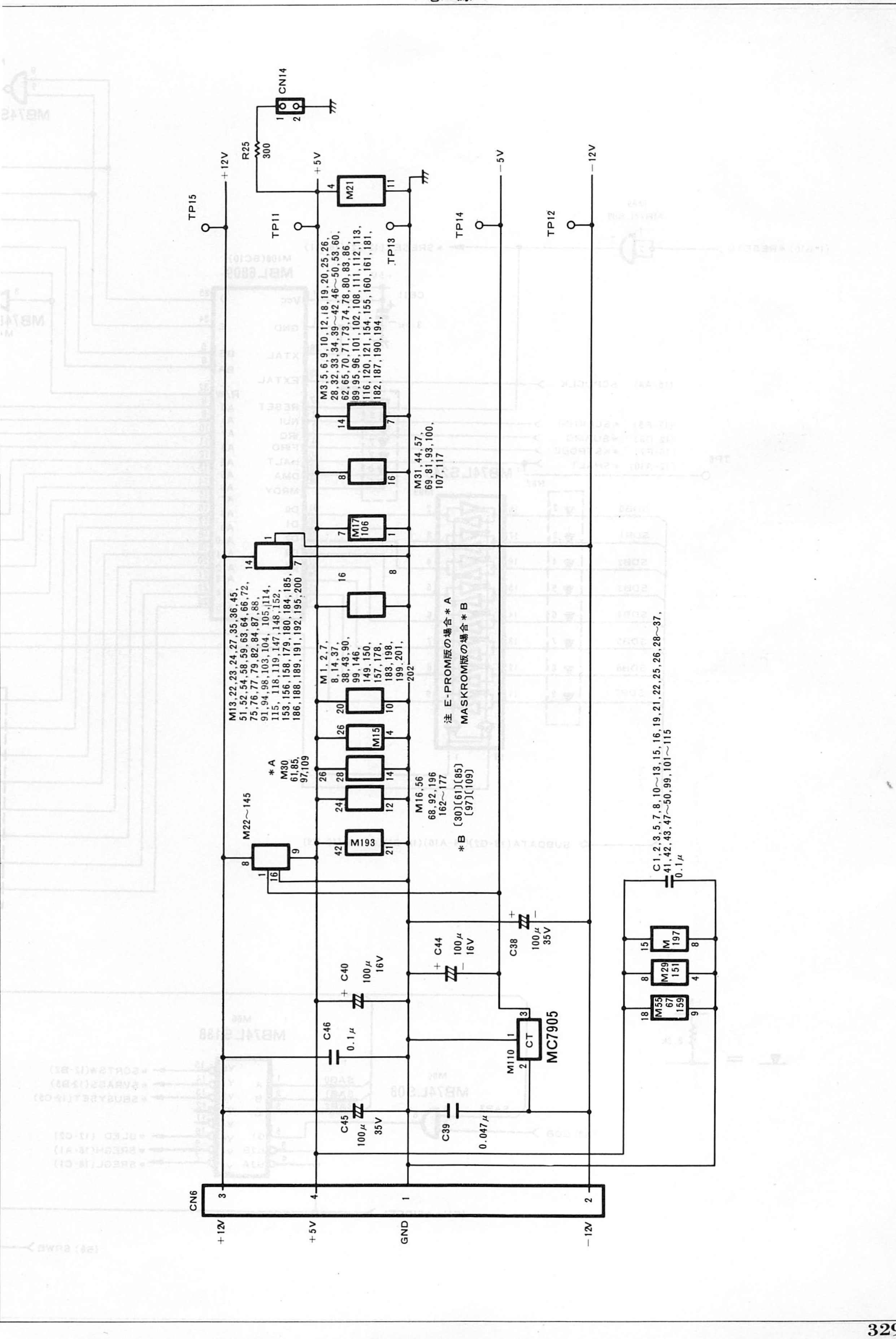
拡張バス/Z80カード・コネクタ



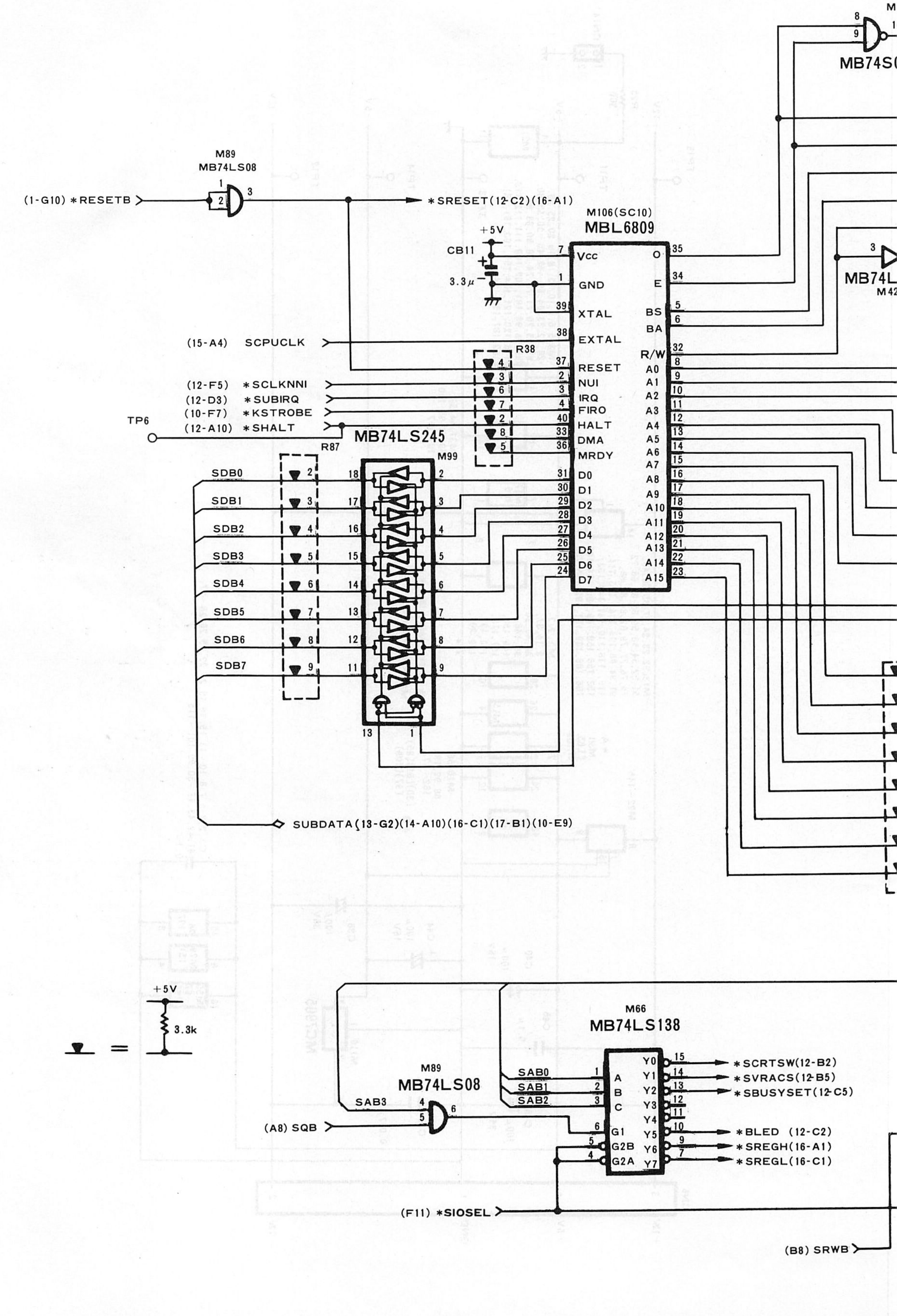
バブル・コネクタ

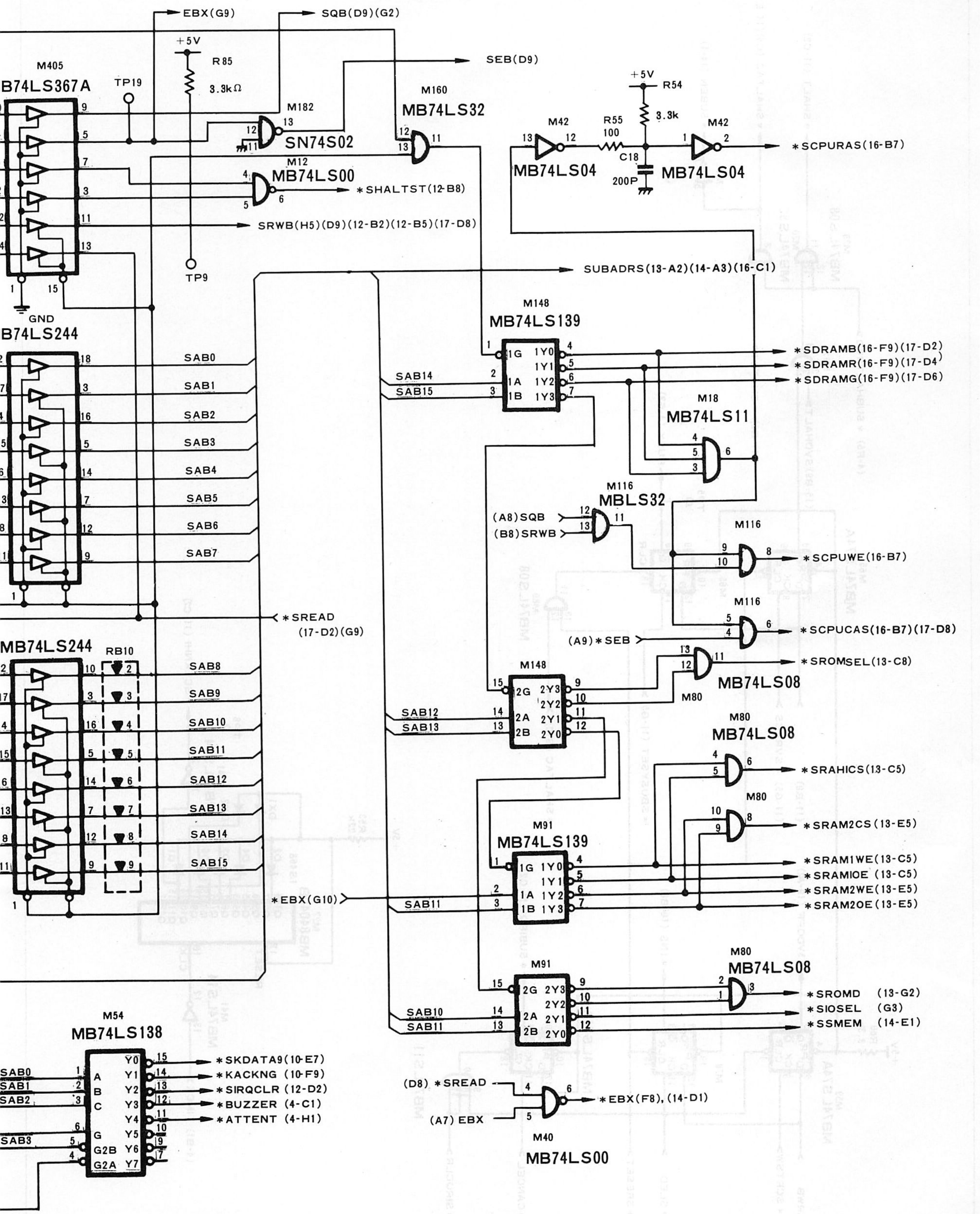


電　源

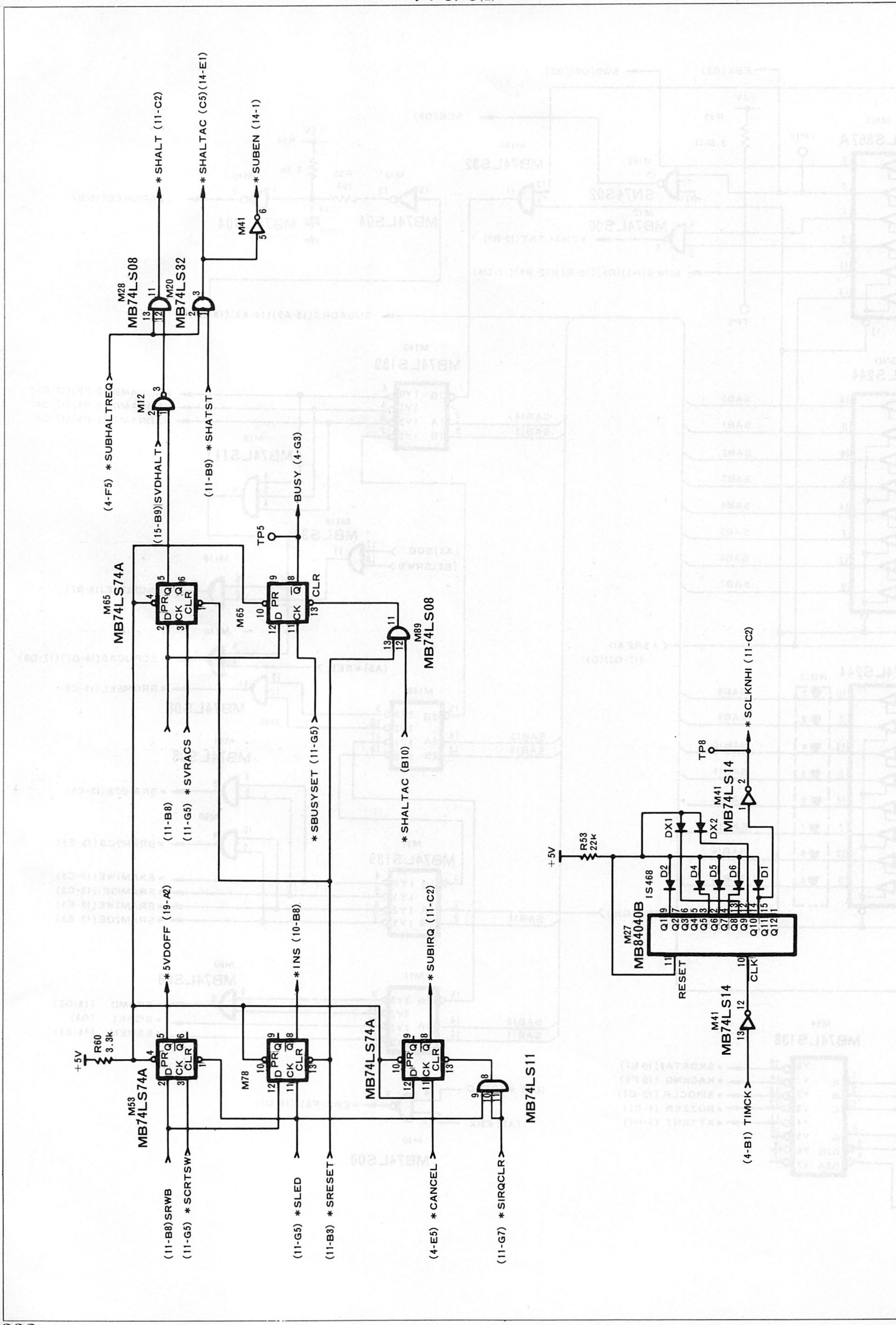

CN14
R25
300
+12V
+5V
−5V
−12V
TP15
TP11
TP13
TP14
TP12
M21
M3, 5, 6, 9, 10, 12, 18, 19, 20, 25, 26, 28, 32, 33, 34, 39〜42, 46〜50, 53, 60, 62, 65, 70, 71, 73, 74, 78, 80, 83, 86, 89, 95, 96, 101, 102, 108, 111, 112, 113, 116, 120, 121, 154, 155, 160, 161, 181, 182, 187, 190, 194,
M31, 44, 57, 69, 81, 93, 100, 107, 117
M17
106
M13, 22, 23, 24, 27, 35, 36, 45, 51, 52, 54, 58, 59, 63, 64, 66, 72, 75, 76, 77, 79, 82, 84, 87, 88, 91, 94, 98, 103, 104, 105, 114, 115, 118, 119, 147, 148, 152, 153, 156, 158, 179, 180, 184, 185, 186, 188, 189, 191, 192, 195, 200
M1, 2, 7, 8, 14, 37, 38, 43, 90, 99, 146, 149, 150, 157, 178, 183, 198, 199, 201, 202
注 E-PROM版の場合＊A
MASKROM版の場合＊B
＊A
M30
61, 85, 97, 109
M15
M22〜145
M193
M16, 56
68, 92, 196
162〜177
＊B (30)(61)(85)
(97)(109)
C1, 2, 3, 5, 7, 8, 10〜13, 15, 16, 19, 21, 22, 25, 26, 28〜37, 41, 42, 43, 47〜50, 99, 101〜115
0.1μ
M
197
M29
151
M55
67
159
C38
100μ
35V
C44
100μ
16V
C40
100μ
16V
C46
0.1μ
MC7905
M110
CT
C45
100μ
35V
C39
0.047μ
CN6
GND

サブCPU(1)


M89
MB74LS08
(1-G10) *RESETB
*SRESET(12-C2)(16-A1)
M106(SC10)
MBL6809
+5V
CB11
3.3μ
Vcc
GND
XTAL
EXTAL
RESET
NUI
IRQ
FIRO
HALT
DMA
MRDY
D0
D1
D2
D3
D4
D5
D6
D7
O
E
BS
BA
R/W
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
(15-A4) SCPUCLK
R38
(12-F5) *SCLKNNI
(12-D3) *SUBIRQ
(10-F7) *KSTROBE
TP6
(12-A10) *SHALT
MB74LS245
R87
M99
SDB0
SDB1
SDB2
SDB3
SDB4
SDB5
SDB6
SDB7
SUBDATA(13-G2)(14-A10)(16-C1)(17-B1)(10-E9)
MB74S
MB74L
M42
+5V
3.3k
M66
MB74LS138
M89
MB74LS08
SAB0
SAB1
SAB2
SAB3
(A8) SQB
A
B
C
G1
G2B
G2A
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
*SCRTSW(12-B2)
*SVRACS(12-B5)
*SBUSYSET(12-C5)
*BLED (12-C2)
*SREGH(16-A1)
*SREGL(16-C1)
(F11) *SIOSEL
(B8) SRWB

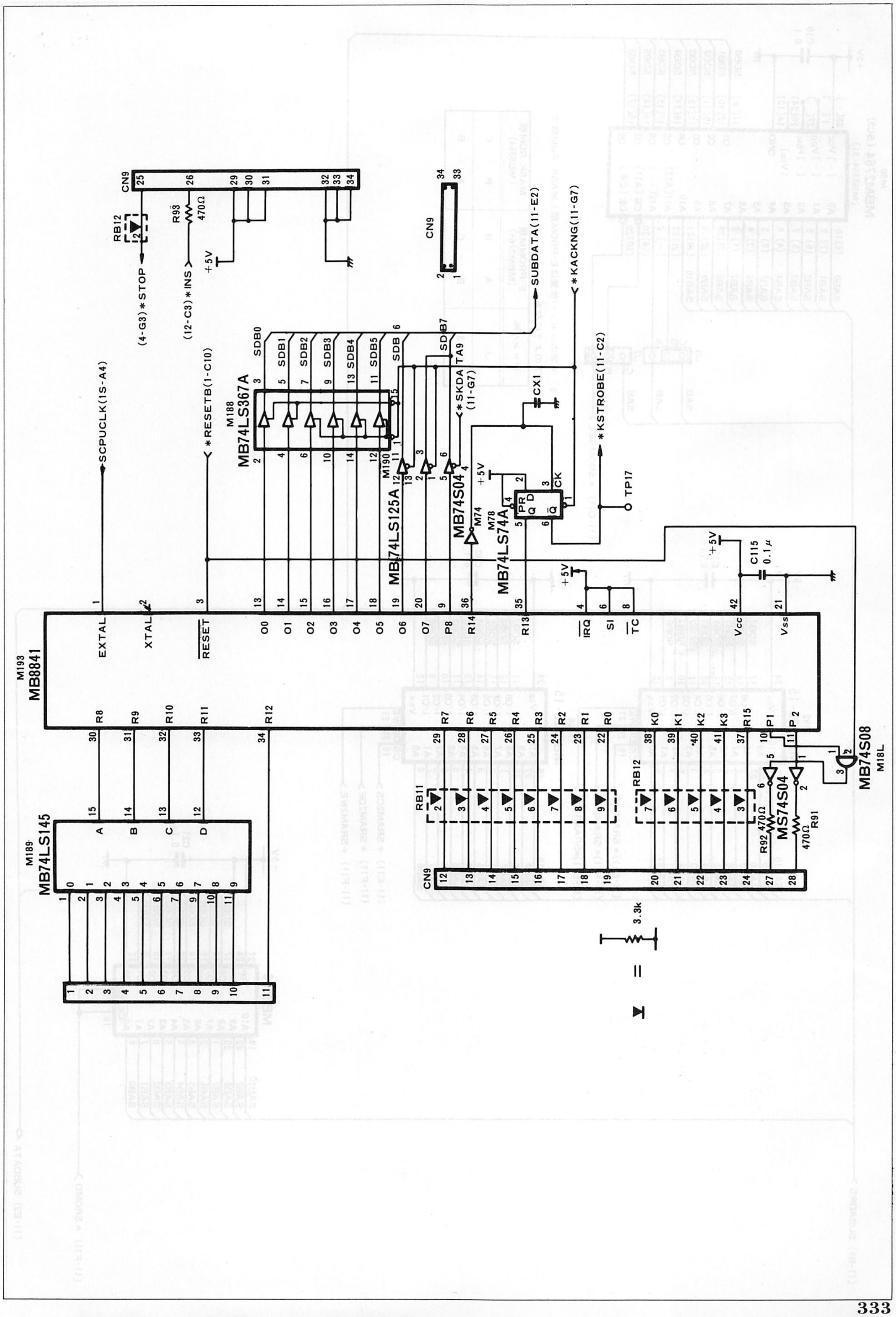

EBX(G9)
SQB(D9)(G2)
SEB(D9)
+5V
R85
3.3kΩ
M405
MB74LS367A
TP19
M182
SN74S02
M160
MB74LS32
M12
MB74LS00
*SHALTST(12-B8)
SRWB(H5)(D9)(12-B2)(12-B5)(17-D8)
TP9
GND
+5V
R54
3.3k
R55
100
C18
200P
M42
MB74LS04
M42
MB74LS04
*SCPURAS(16-B7)
SUBADRS(13-A2)(14-A3)(16-C1)
MB74LS244
SAB0
SAB1
SAB2
SAB3
SAB4
SAB5
SAB6
SAB7
M148
MB74LS139
SAB14
SAB15
*SDRAMB(16-F9)(17-D2)
*SDRAMR(16-F9)(17-D4)
*SDRAMG(16-F9)(17-D6)
M18
MB74LS11
M116
MBLS32
(A8)SQB
(B8)SRWB
M116
*SCPUWE(16-B7)
M116
*SCPUCAS(16-B7)(17-D8)
(A9)*SEB
*SROMSEL(13-C8)
MB74LS08
M80
*SREAD
(17-D2)(G9)
MB74LS244
RB10
SAB8
SAB9
SAB10
SAB11
SAB12
SAB13
SAB14
SAB15
M148
SAB12
SAB13
M80
MB74LS08
*SRAHICS(13-C5)
M80
*SRAM2CS(13-E5)
M91
MB74LS139
*SRAM1WE(13-C5)
*SRAMIOE (13-C5)
*SRAM2WE(13-E5)
*SRAM2OE(13-E5)
*EBX(G10)
SAB11
M91
M80
MB74LS08
SAB10
SAB11
*SROMD (13-G2)
*SIOSEL (G3)
*SSMEM (14-E1)
M54
MB74LS138
SAB0
SAB1
SAB2
SAB3
*SKDATA9(10-E7)
*KACKNG (10-F9)
*SIRQCLR (12-D2)
*BUZZER (4-C1)
*ATTENT (4-H1)
(D8) *SREAD
(A7) EBX
*EBX(F8), (14-D1)
M40
MB74LS00

サブCPU(2)

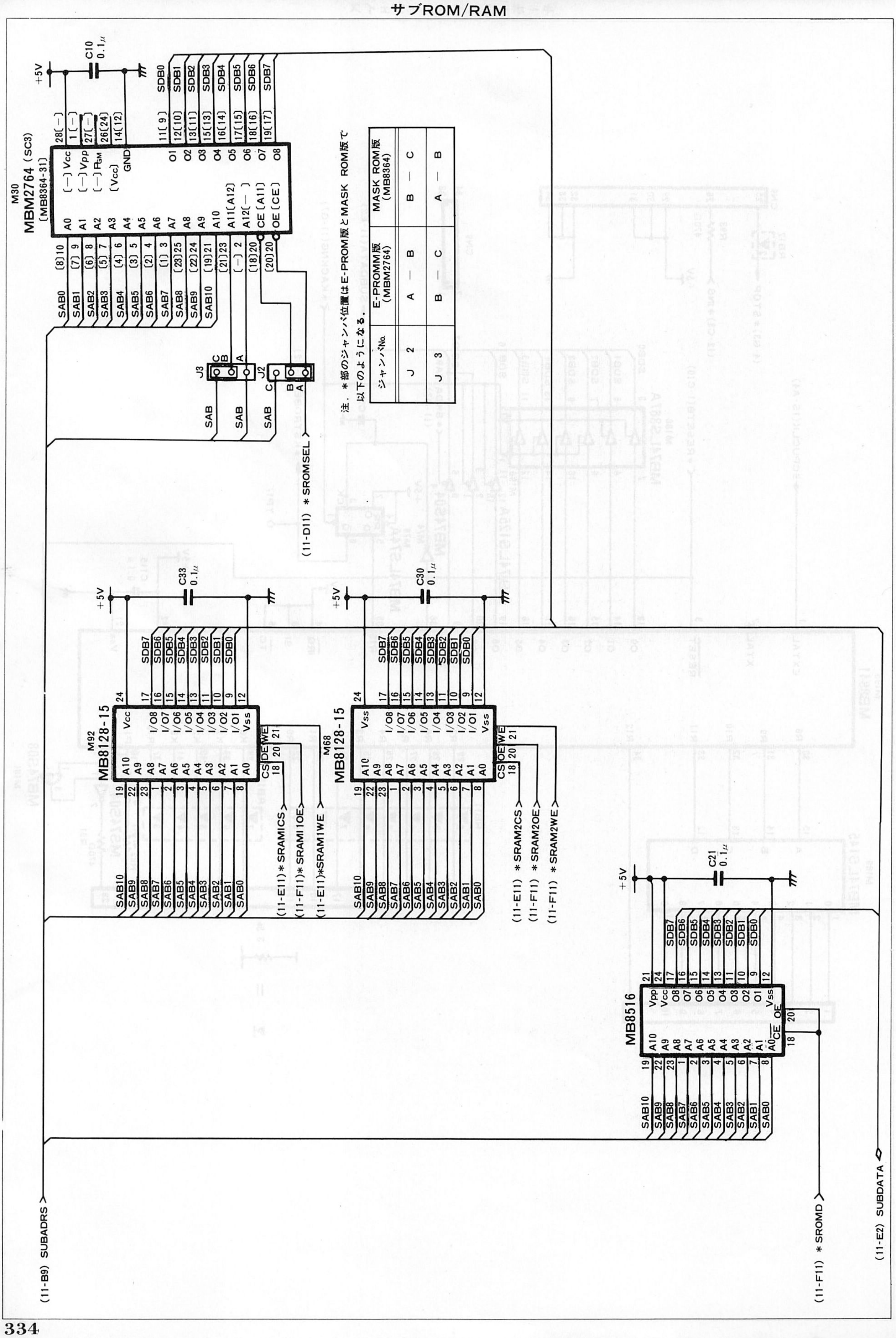


＊SHALT (11-C2)
＊SHALTAC (C5)(14-E1)
＊SUBEN (14-1)
M41
M28
MB74LS08
M20
MB74LS32
M12
(4-F5) ＊SUBHALTREQ
(15-B9)SVDHALT
(11-B9) ＊SHATST
BUSY (4-G3)
TP5
M65
MB74LS74A
M89
MB74LS08
(11-B8)
(11-G5) ＊SVRACS
＊SBUSYSET (11-G5)
＊SHALTAC (B10)
＊5VDOFF (19-A2)
＊INS (10-B8)
＊SUBIRQ (11-C2)
+5V
R60
3.3k
M53
MB74LS74A
M78
MB74LS11
(11-B8)SRWB
(11-G5) ＊SCRTSW
(11-G5) ＊SLED
(11-B3) ＊SRESET
(4-E5) ＊CANCEL
(11-G7) ＊SIRQCLR
＊SCLKNHI (11-C2)
TP8
M41
MB74LS14
DX1
DX2
R53
22k
IS468
D2
D4
D5
D6
D1
M27
MB8040B
RESET
CLK
(4-B1) TIMCK

キーボード・インターフェイス



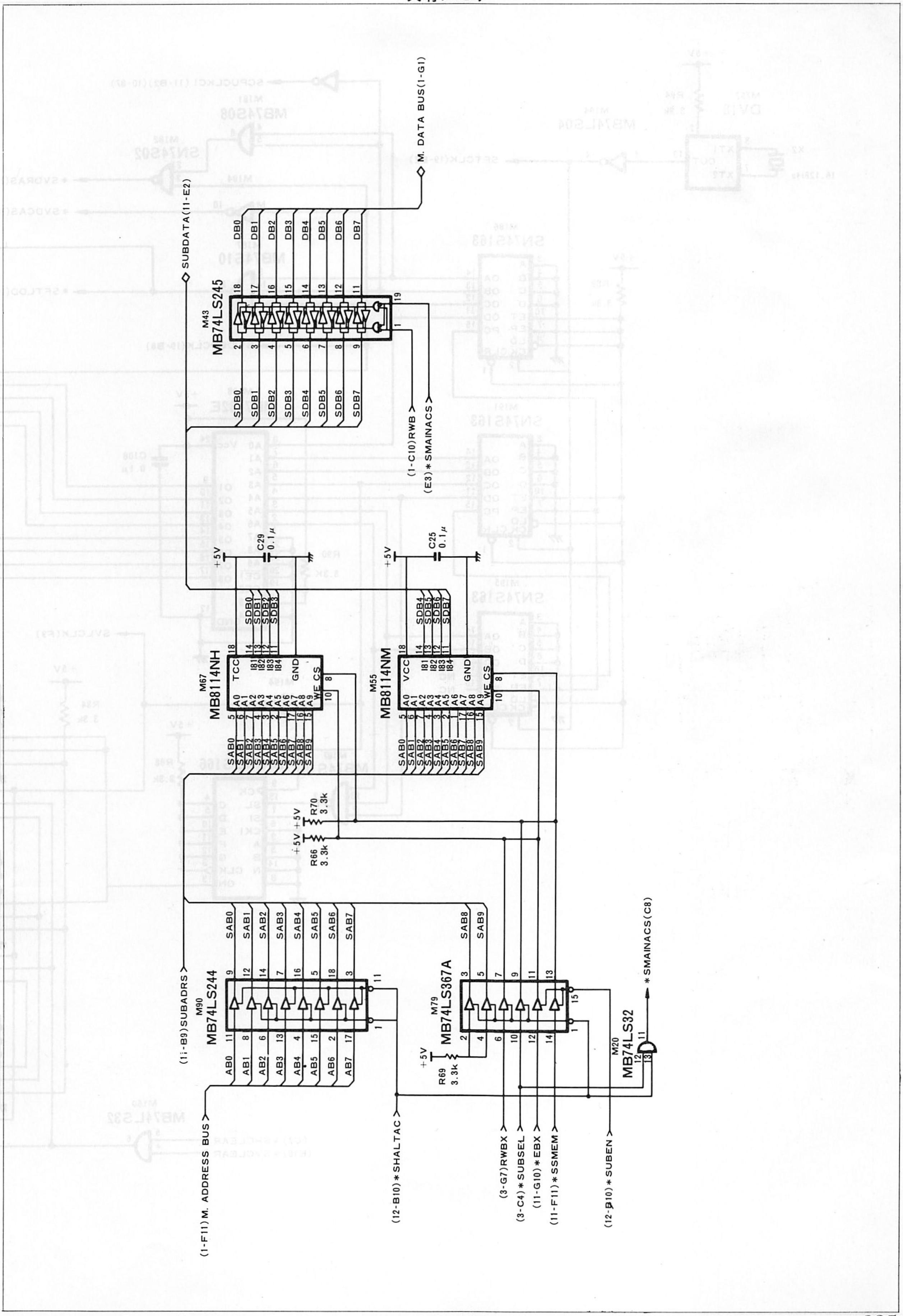
サブROM/RAM



注．＊部のジャンパ位置はE-PROM版とMASK ROM版で以下のようになる。

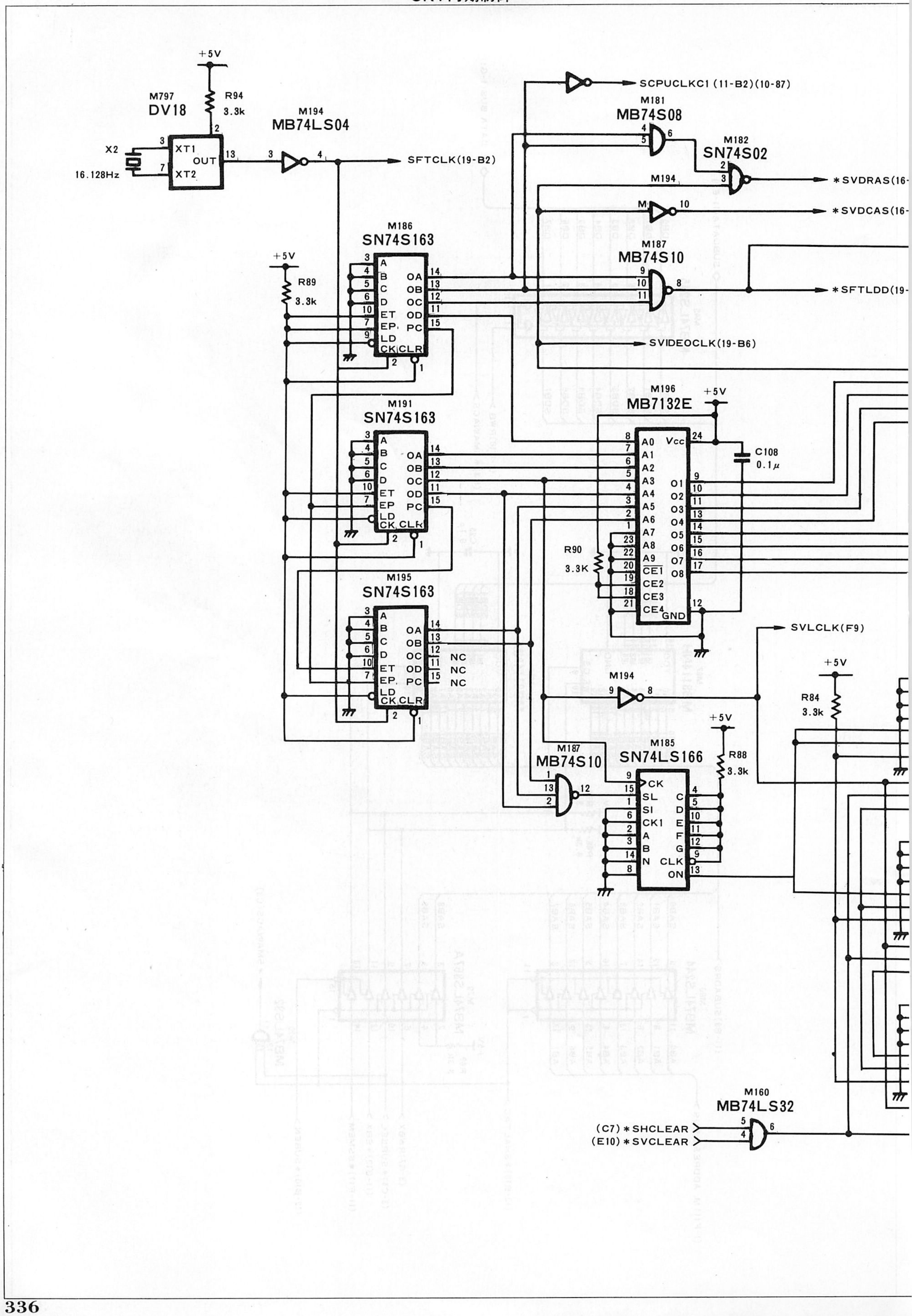
ジャンパNo.	E-PROMM版 (MBM2764)	MASK ROM版 (MB8364)
J 2	A — B	B — C
J 3	B — C	A — B

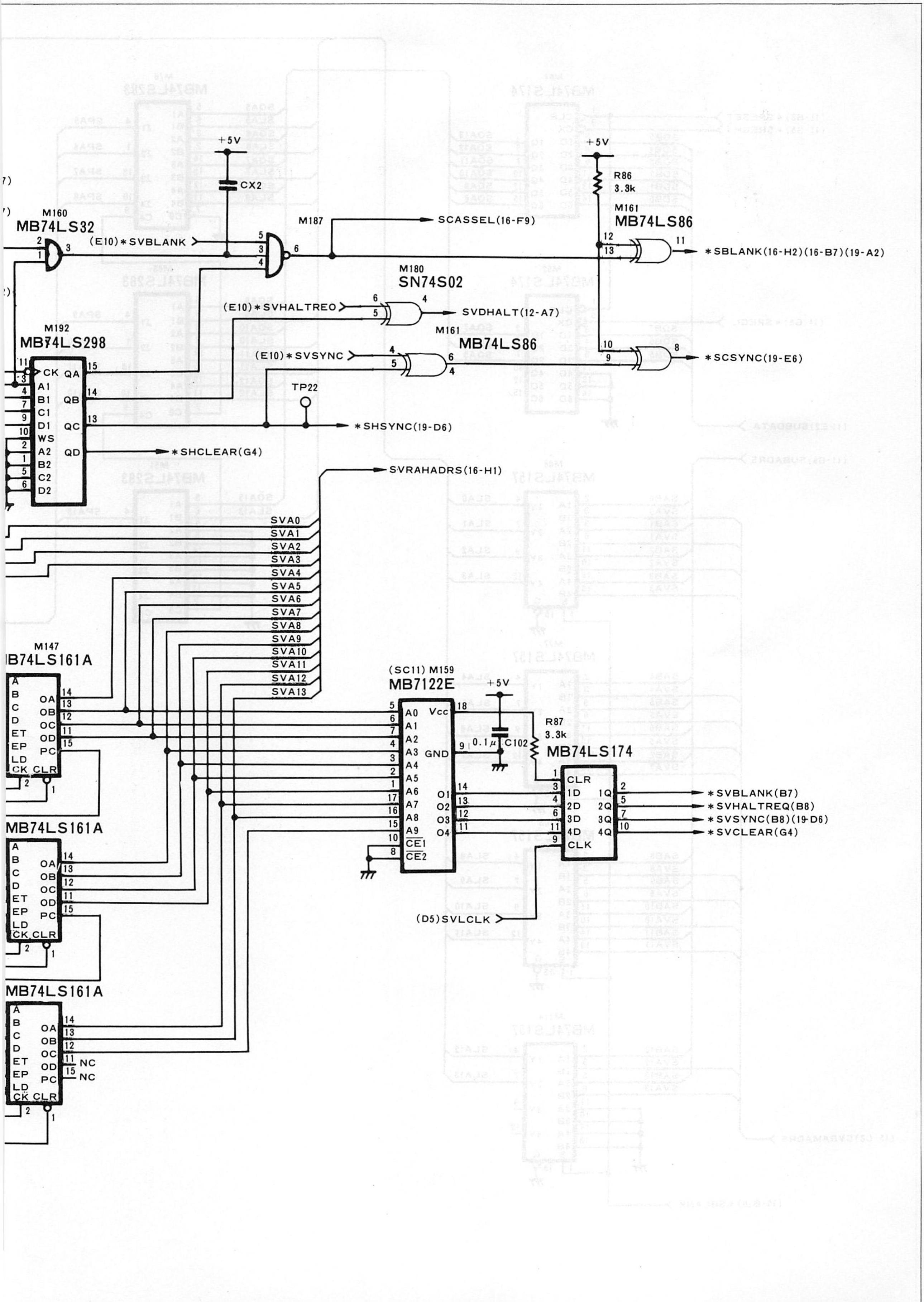
共有メモリ



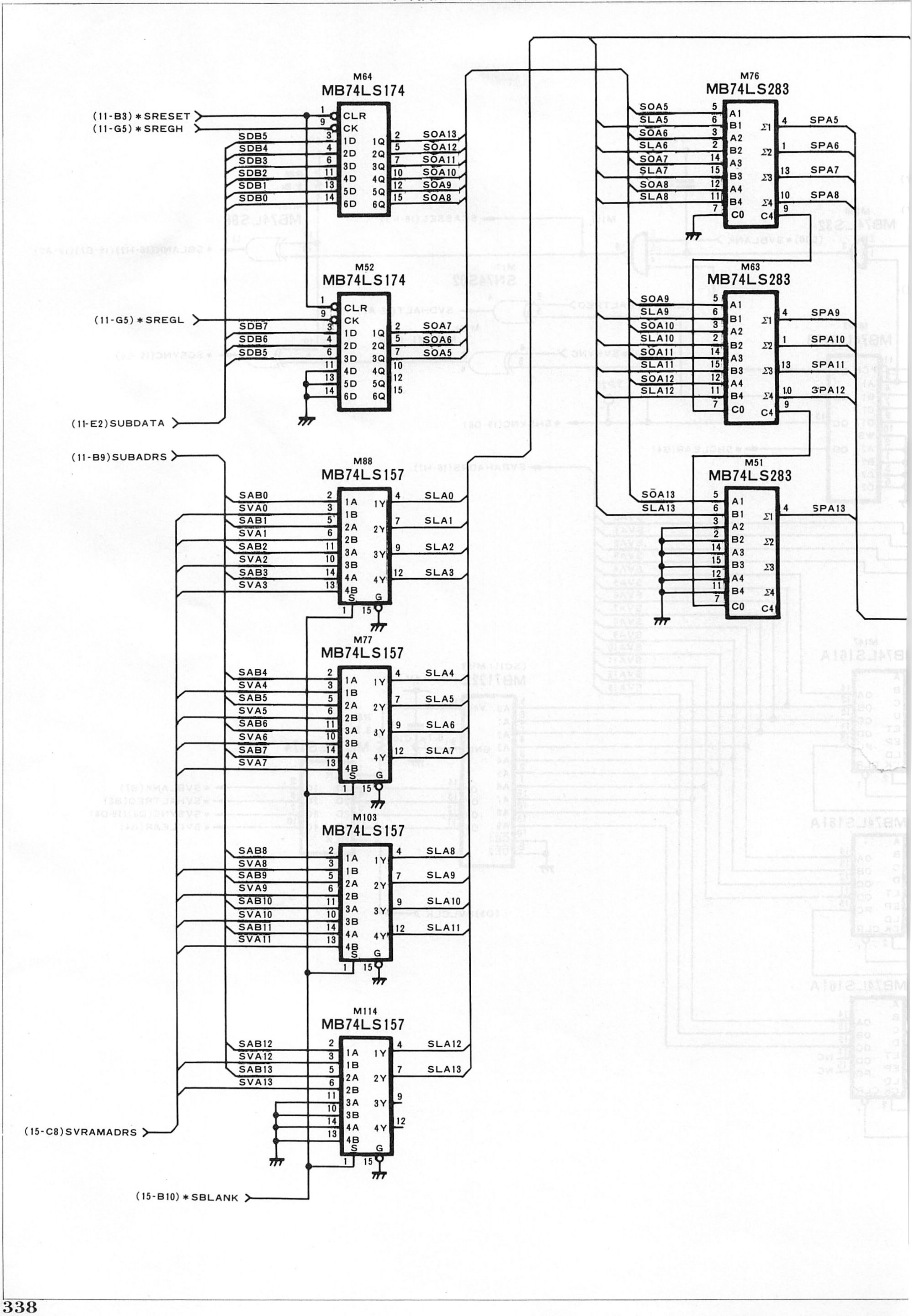
M. DATA BUS(1-G1)
SUBDATA(11-E2)
M43
MB74LS245
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
SDB0
SDB1
SDB2
SDB3
SDB4
SDB5
SDB6
SDB7
(1-C10)RWB
(E3)＊SMAINACS
C29
0.1μ
C25
0.1μ
+5V
M67
MB8114NH
M55
MB8114NM
SAB0
SAB1
SAB2
SAB3
SAB4
SAB5
SAB6
SAB7
SAB8
SAB9
R70
3.3k
R66
3.3k
(1i-B9)SUBADRS
M90
MB74LS244
AB0
AB1
AB2
AB3
AB4
AB5
AB6
AB7
M79
MB74LS367A
R69
3.3k
M20
MB74LS32
＊SMAINACS(C8)
(1-F11) M. ADDRESS BUS
(12-B10)＊SHALTAC
(3-G7)RWBX
(3-C4)＊SUBSEL
(11-G10)＊EBX
(11-F11)＊SSMEM
(12-B10)＊SUBEN

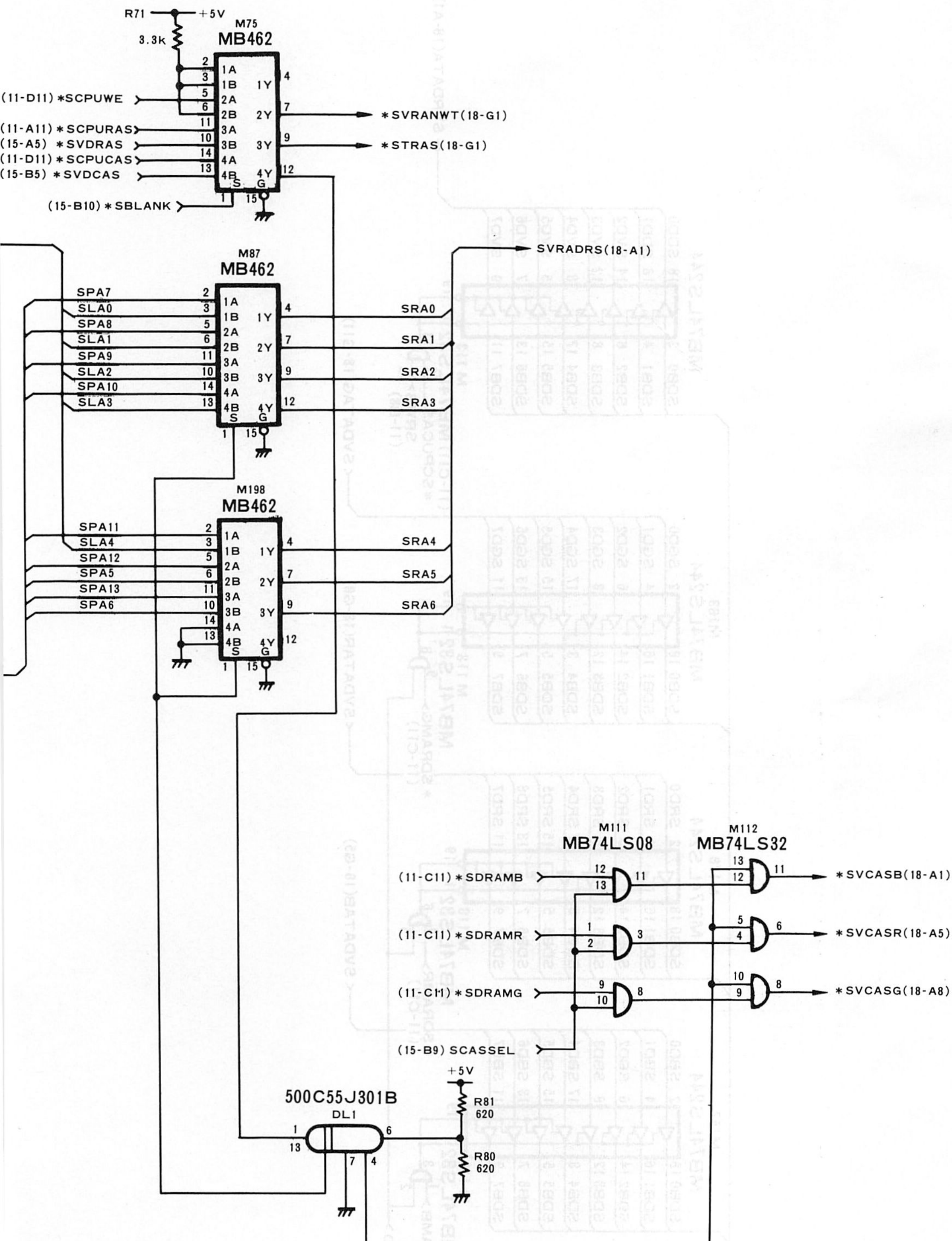
CRT同期制御


+5V
M797
DV18
R94
3.3k
X2
16.128Hz
XT1
XT2
OUT
M194
MB74LS04
SFTCLK(19-B2)
SCPUCLKC1 (11-B2)(10-87)
M181
MB74S08
M182
SN74S02
*SVDRAS(16-
*SVDCAS(16-
M186
SN74S163
M187
MB74S10
*SFTLDD(19-
SVIDEOCLK(19-B6)
R89
3.3k
M191
SN74S163
M196
MB7132E
C108
0.1μ
R90
3.3K
M195
SN74S163
NC
SVLCLK(F9)
R84
3.3k
R88
3.3k
M185
SN74LS166
M160
MB74LS32
(C7) *SHCLEAR
(E10) *SVCLEAR

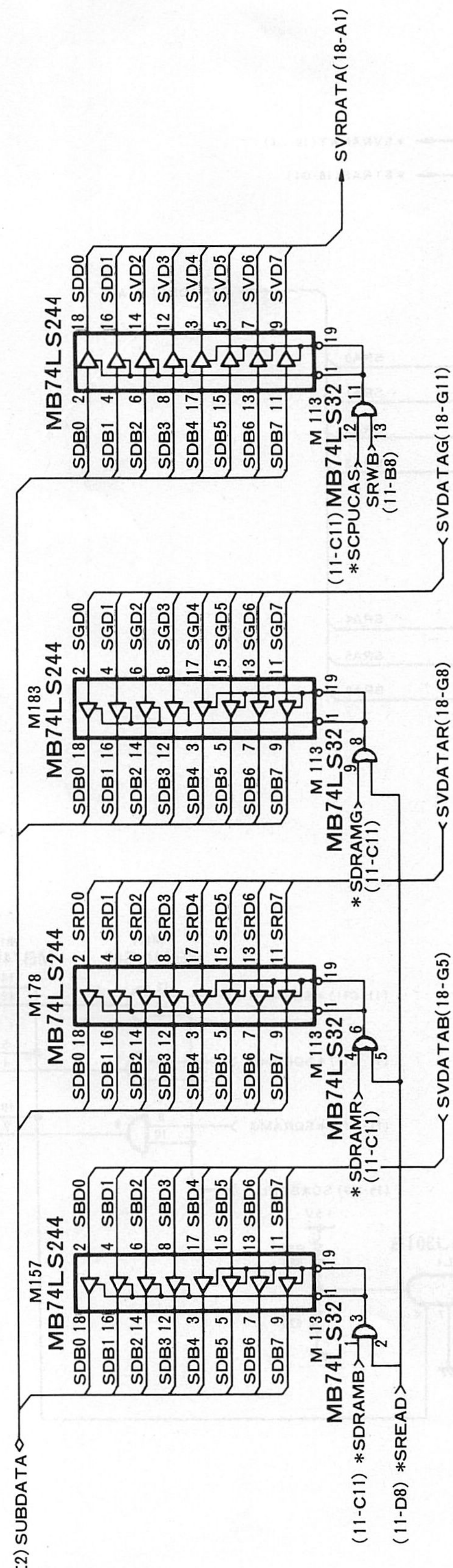

M160
MB74LS32
(E10)＊SVBLANK
+5V
CX2
M187
SCASSEL(16-F9)
+5V
R86
3.3k
M161
MB74LS86
＊SBLANK(16-H2)(16-B7)(19-A2)
M180
SN74S02
(E10)＊SVHALTREQ
SVDHALT(12-A7)
M161
MB74LS86
(E10)＊SVSYNC
＊SCSYNC(19-E6)
M192
MB74LS298
TP22
＊SHSYNC(19-D6)
＊SHCLEAR(G4)
SVRAHADRS(16-H1)
SVA0
SVA1
SVA2
SVA3
SVA4
SVA5
SVA6
SVA7
SVA8
SVA9
SVA10
SVA11
SVA12
SVA13
M147
MB74LS161A
(SC11) M159
MB7122E
+5V
R87
3.3k
0.1μ C102
MB74LS174
＊SVBLANK(B7)
＊SVHALTREQ(B8)
＊SVSYNC(B8)(19-D6)
＊SVCLEAR(G4)
MB74LS161A
(D5)SVLCLK
MB74LS161A
NC
NC

V-RAMバッファ(1)

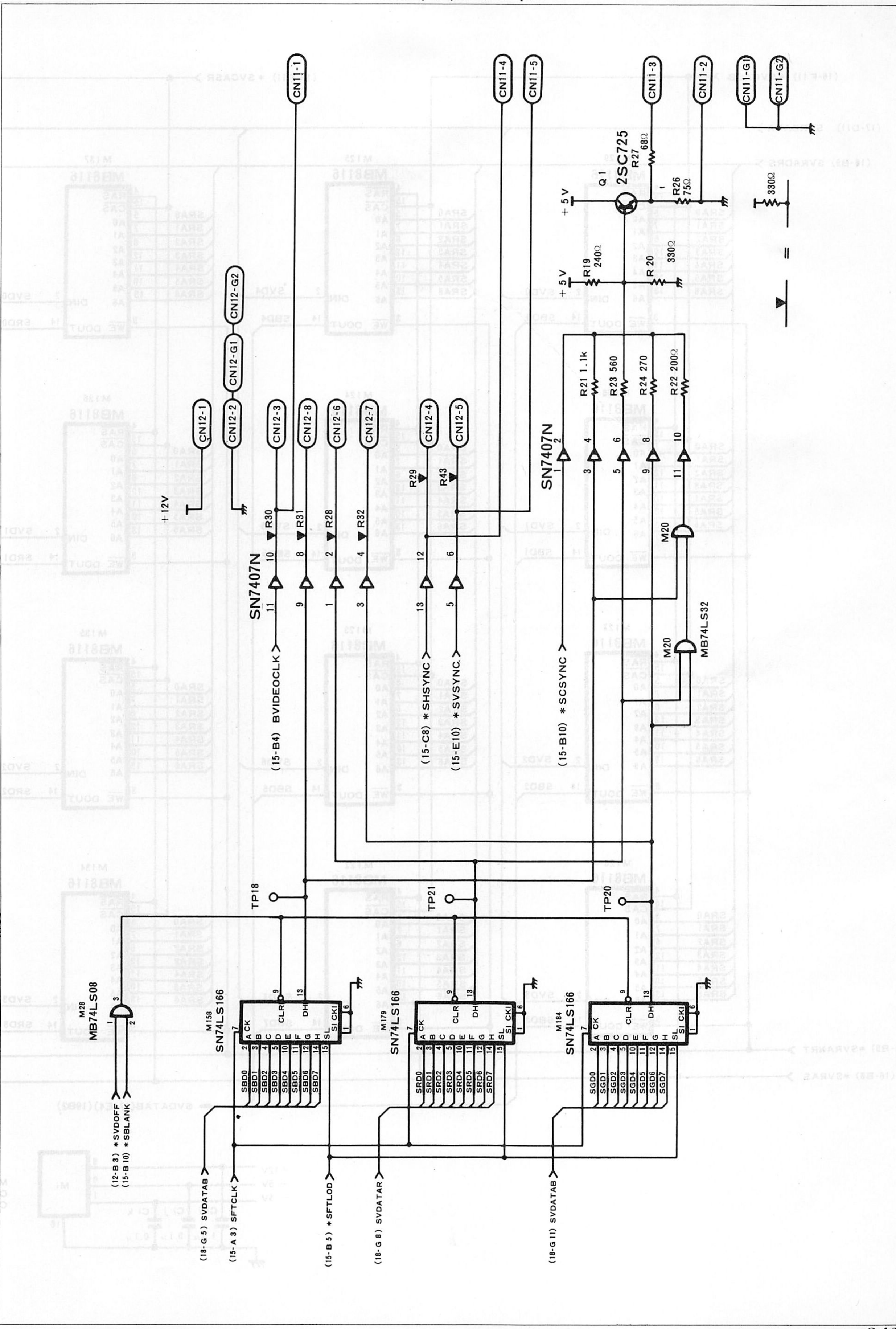

M64
MB74LS174
(11-B3) *SRESET
(11-G5) *SREGH
SDB5
SDB4
SDB3
SDB2
SDB1
SDB0
SOA13
SOA12
SOA11
SOA10
SOA9
SOA8
M52
MB74LS174
(11-G5) *SREGL
SDB7
SDB6
SDB5
SOA7
SOA6
SOA5
(11-E2)SUBDATA
(11-B9)SUBADRS
M88
MB74LS157
SAB0
SVA0
SAB1
SVA1
SAB2
SVA2
SAB3
SVA3
SLA0
SLA1
SLA2
SLA3
M77
MB74LS157
SAB4
SVA4
SAB5
SVA5
SAB6
SVA6
SAB7
SVA7
SLA4
SLA5
SLA6
SLA7
M103
MB74LS157
SAB8
SVA8
SAB9
SVA9
SAB10
SVA10
SAB11
SVA11
SLA8
SLA9
SLA10
SLA11
M114
MB74LS157
SAB12
SVA12
SAB13
SVA13
SLA12
SLA13
(15-C8)SVRAMADRS
(15-B10) *SBLANK
M76
MB74LS283
SOA5
SLA5
SOA6
SLA6
SOA7
SLA7
SOA8
SLA8
SPA5
SPA6
SPA7
SPA8
M63
MB74LS283
SOA9
SLA9
SOA10
SLA10
SOA11
SLA11
SOA12
SLA12
SPA9
SPA10
SPA11
SPA12
M51
MB74LS283
SOA13
SLA13
SPA13


R71
+5V
3.3k
M75
MB462
(11-D11) *SCPUWE
(11-A11) *SCPURAS
(15-A5) *SVDRAS
(11-D11) *SCPUCAS
(15-B5) *SVDCAS
(15-B10) *SBLANK
*SVRANWT (18-G1)
*STRAS (18-G1)
M87
MB462
SPA7
SLA0
SPA8
SLA1
SPA9
SLA2
SPA10
SLA3
SVRADRS (18-A1)
SRA0
SRA1
SRA2
SRA3
M198
MB462
SPA11
SLA4
SPA12
SPA5
SPA13
SPA6
SRA4
SRA5
SRA6
M111
MB74LS08
M112
MB74LS32
(11-C11) *SDRAMB
(11-C11) *SDRAMR
(11-C11) *SDRAMG
(15-B9) SCASSEL
*SVCASB (18-A1)
*SVCASR (18-A5)
*SVCASG (18-A8)
500C55J301B
DL1
+5V
R81
620
R80
620

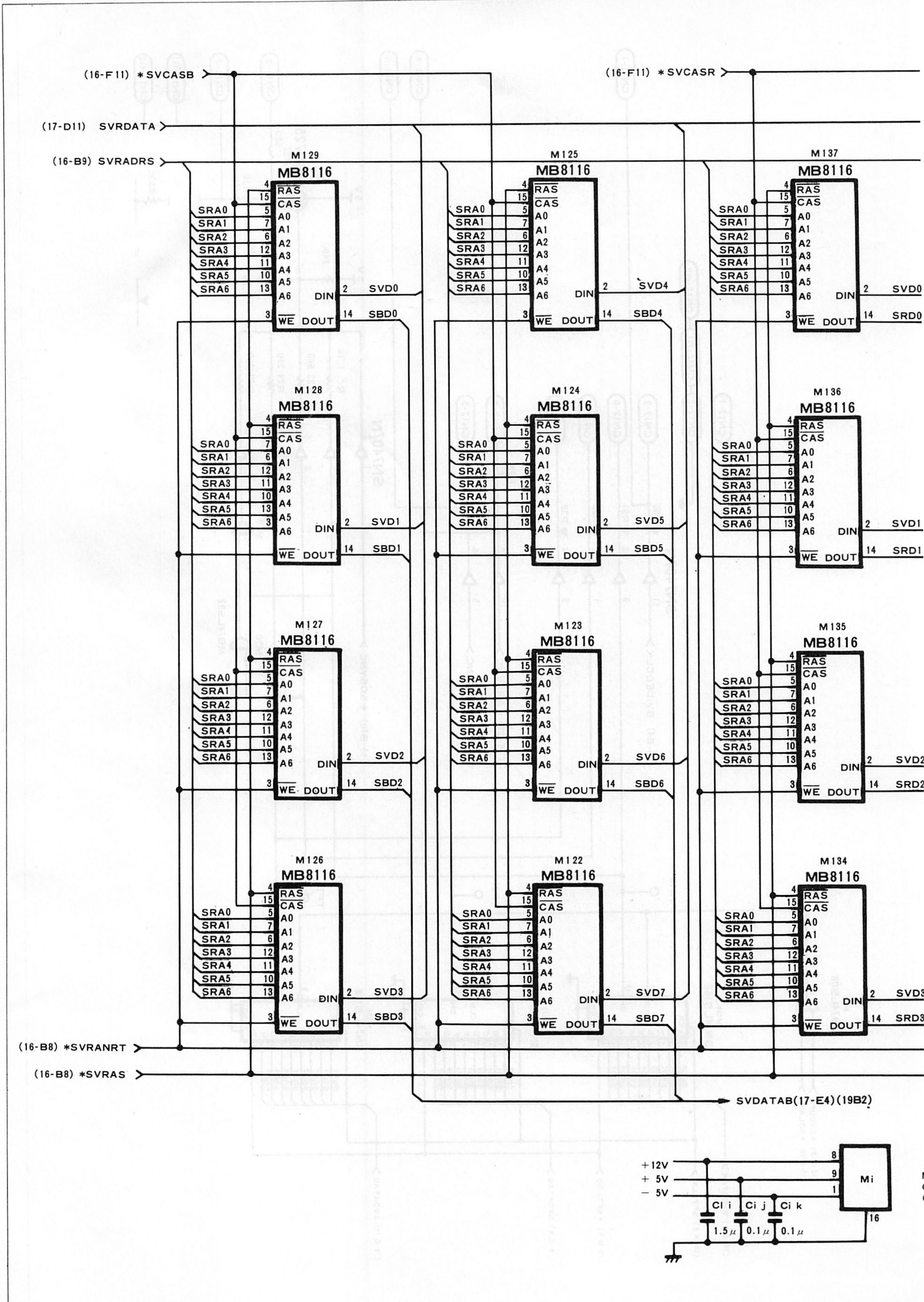
V-RAMバッファ(2)

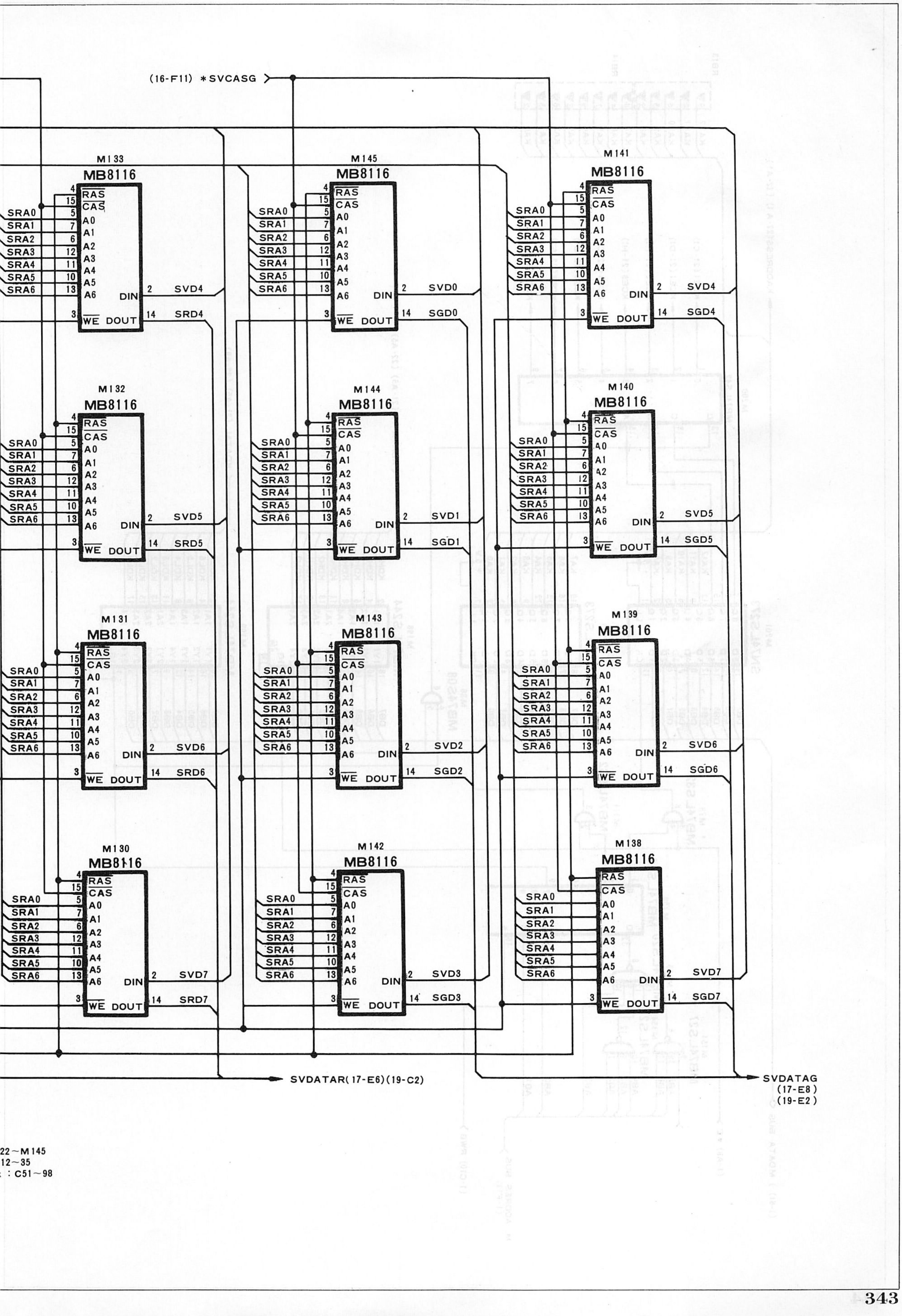

SVRDATA(18-A1)
MB74LS244
SDD0 SDD1 SVD2 SVD3 SVD4 SVD5 SVD6 SVD7
SDB0 SDB1 SDB2 SDB3 SDB4 SDB5 SDB6 SDB7
M 113
MB74LS32
(11-C11) *SCPUCAS
SRWB
(11-B8)
SVDATAG(18-G11)
M183
MB74LS244
SGD0 SGD1 SGD2 SGD3 SGD4 SGD5 SGD6 SGD7
*SDRAMG
(11-C11)
SVDATAR(18-G8)
M178
MB74LS244
SRD0 SRD1 SRD2 SRD3 SRD4 SRD5 SRD6 SRD7
*SDRAMR
(11-C11)
SVDATAB(18-G5)
M157
MB74LS244
SBD0 SBD1 SBD2 SBD3 SBD4 SBD5 SBD6 SBD7
(11-E2) SUBDATA
(11-C11) *SDRAMB
(11-D8) *SREAD

CRTインターフェイス

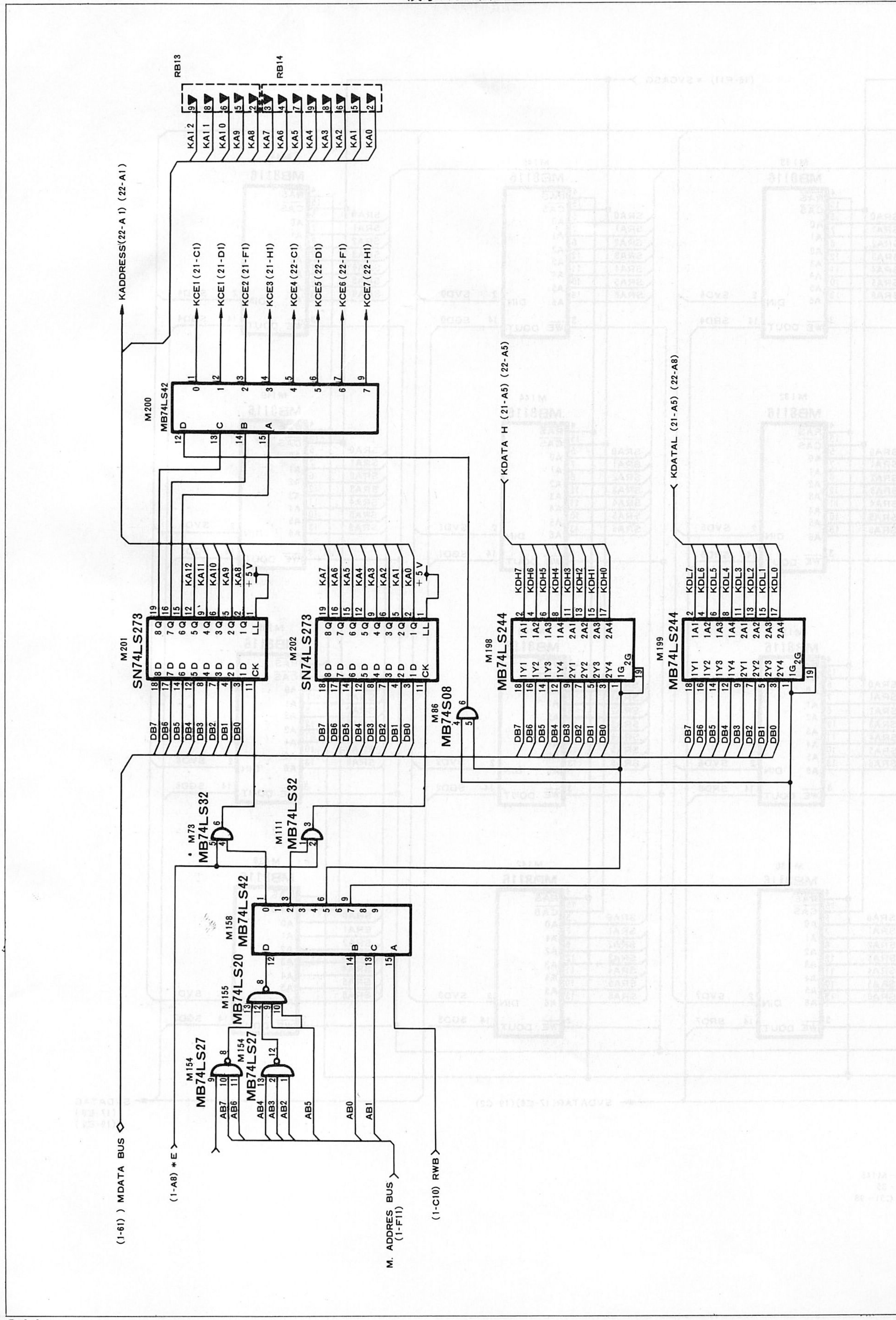


V-RAM

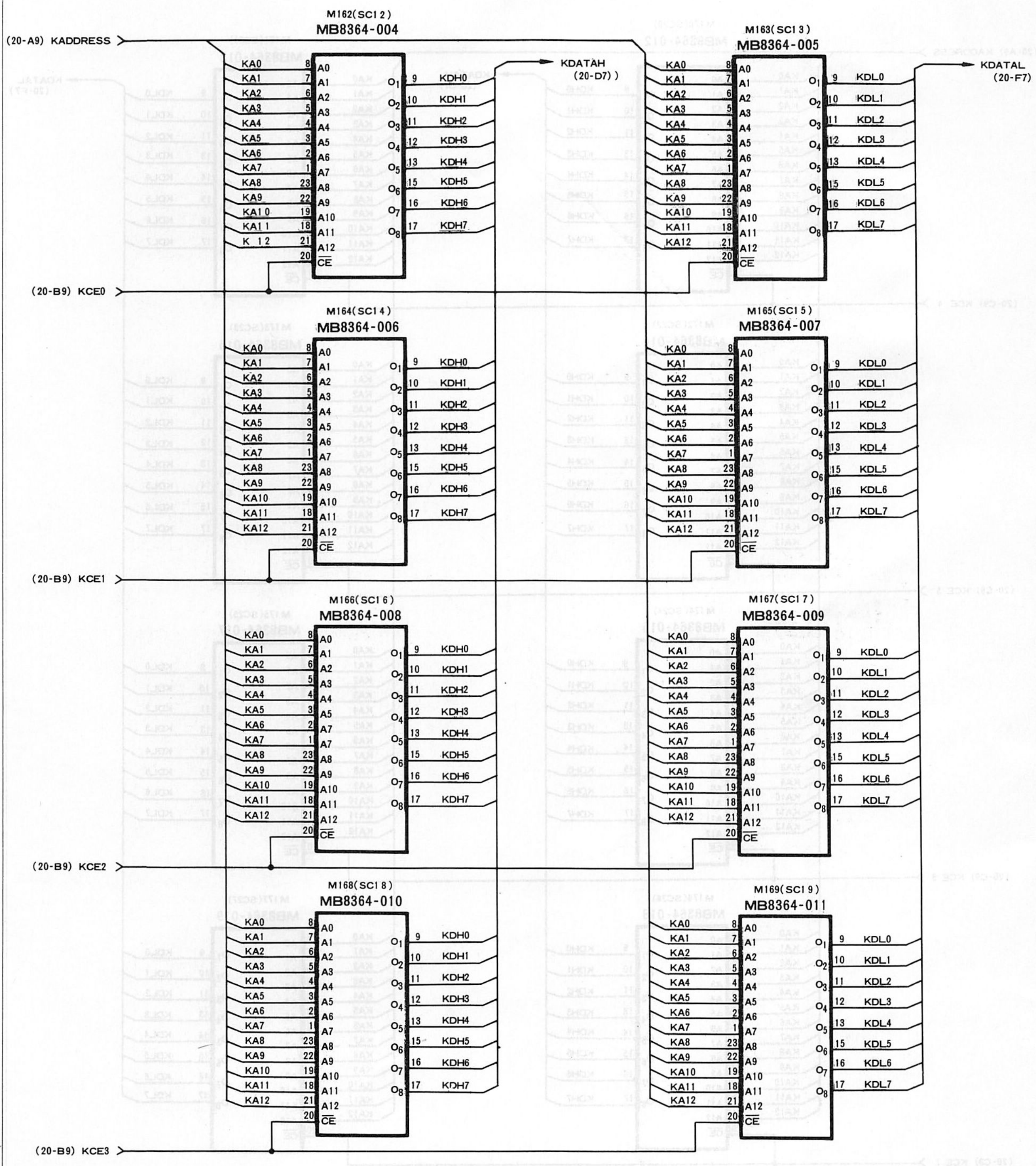

(16-F11) *SVCASB
(16-F11) *SVCASR
(17-D11) SVRDATA
(16-B9) SVRADRS
M129
M128
M127
M126
M125
M124
M123
M122
M137
M136
M135
M134
MB8116
RAS
CAS
A0
A1
A2
A3
A4
A5
A6
DIN
WE DOUT
SRA0
SRA1
SRA2
SRA3
SRA4
SRA5
SRA6
SVD0
SVD1
SVD2
SVD3
SVD4
SVD5
SVD6
SVD7
SBD0
SBD1
SBD2
SBD3
SBD4
SBD5
SBD6
SBD7
SRD0
SRD1
SRD2
SRD3
(16-B8) *SVRANRT
(16-B8) *SVRAS
SVDATAB(17-E4)(19B2)
+12V
+ 5V
− 5V
Ci i
Ci j
Ci k
1.5μ
0.1μ
0.1μ
Mi


(16-F11) *SVCASG
M133
MB8116
M145
MB8116
M141
MB8116
M132
MB8116
M144
MB8116
M140
MB8116
M131
MB8116
M143
MB8116
M139
MB8116
M130
MB8116
M142
MB8116
M138
MB8116
RAS
CAS
A0
A1
A2
A3
A4
A5
A6
DIN
WE DOUT
SRA0
SRA1
SRA2
SRA3
SRA4
SRA5
SRA6
SVD0
SVD1
SVD2
SVD3
SVD4
SVD5
SVD6
SVD7
SRD4
SRD5
SRD6
SRD7
SGD0
SGD1
SGD2
SGD3
SGD4
SGD5
SGD6
SGD7
SVDATAR(17-E6)(19-C2)
SVDATAG
(17-E8)
(19-E2)
22～M145
12～35
: C51～98

漢字ROM(1)

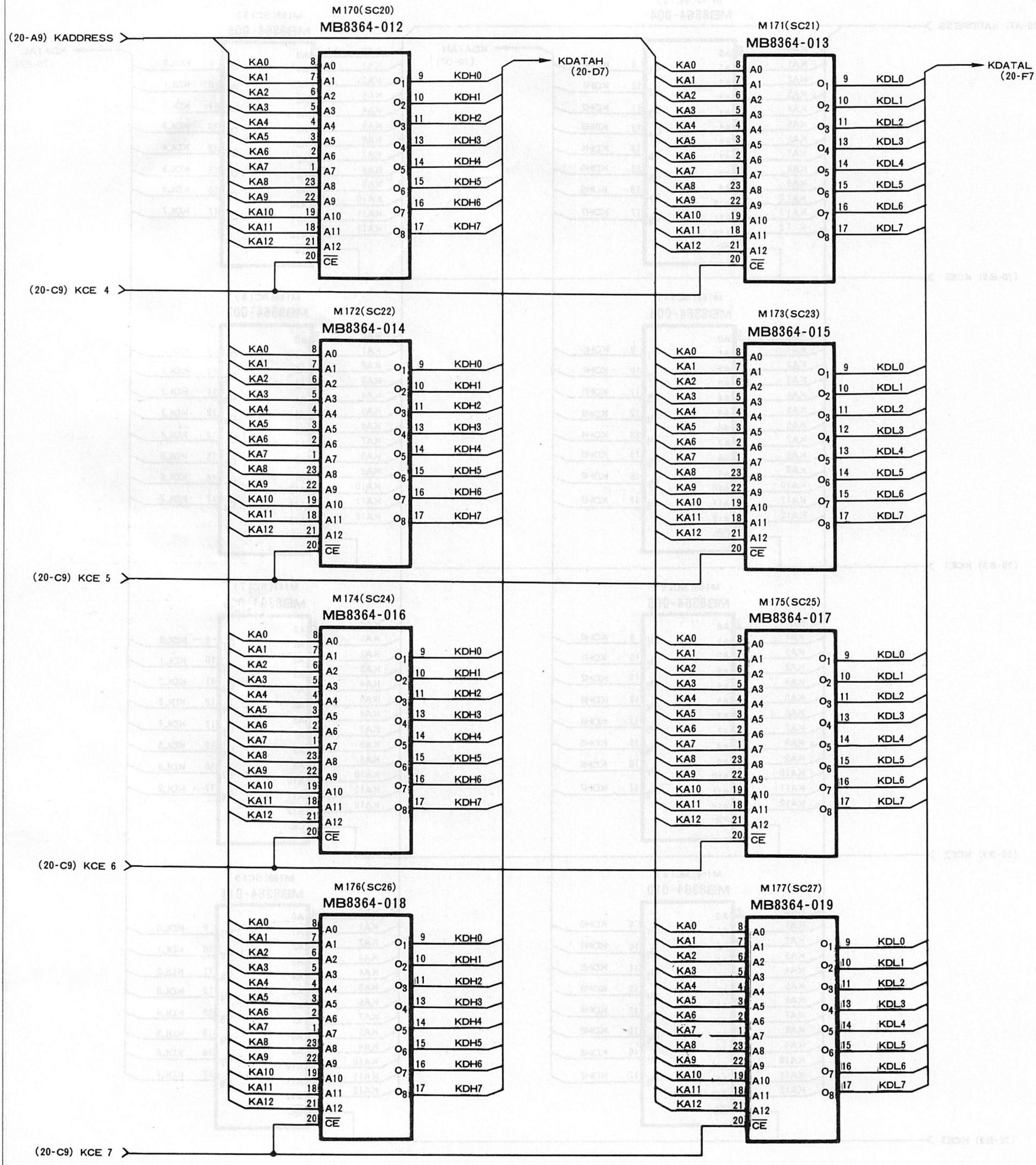

RB13
RB14
KA12 KA11 KA10 KA9 KA8 KA7 KA6 KA5 KA4 KA3 KA2 KA1 KA0
KADDRESS(22-A1) (22-A1)
KCE1 (21-C1)
KCE1 (21-D1)
KCE2 (21-F1)
KCE3 (21-H1)
KCE4 (22-C1)
KCE5 (22-D1)
KCE6 (22-F1)
KCE7 (22-H1)
M200
MB74LS42
M201
SN74LS273
M202
SN74LS273
+5V
M198
MB74LS244
M199
MB74LS244
KDATA H (21-A5) (22-A5)
KDATAL (21-A5) (22-A8)
KDH7 KDH6 KDH5 KDH4 KDH3 KDH2 KDH1 KDH0
KDL7 KDL6 KDL5 KDL4 KDL3 KDL2 KDL1 KDL0
DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0
M86
MB74S08
M73
MB74LS32
M111
MB74LS32
M158
MB74LS42
M155
MB74LS20
M154
MB74LS27
AB7 AB6 AB4 AB3 AB2 AB5 AB0 AB1
(1-61) MDATA BUS
(1-A8) *E
M. ADDRES BUS (1-F11)
(1-C10) RWB

漢字ROM(2)



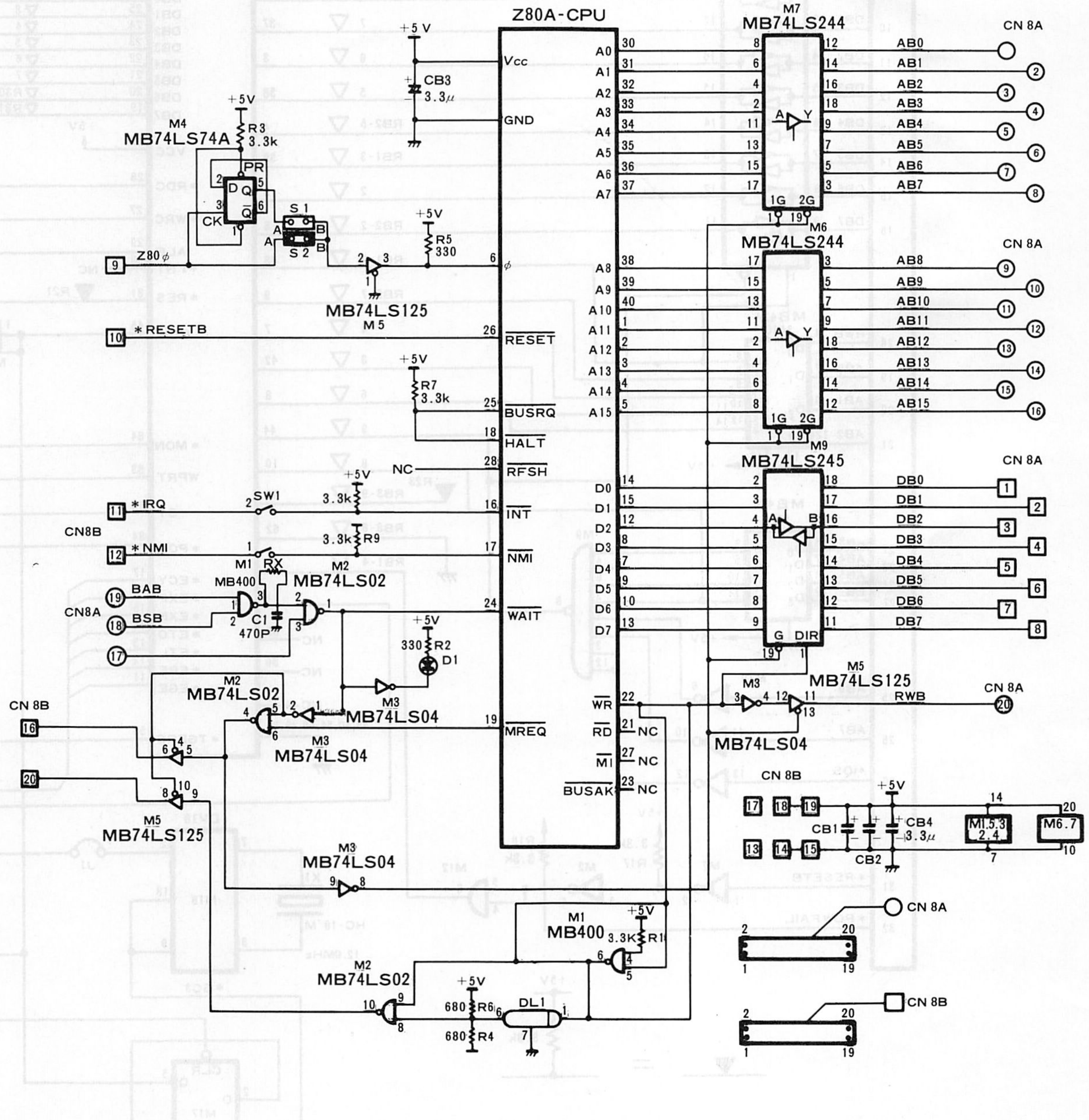
(20-A9) KADDRESS
(20-B9) KCE0
(20-B9) KCE1
(20-B9) KCE2
(20-B9) KCE3
KDATAH (20-D7)
KDATAL (20-F7)
M162(SC12) MB8364-004
M163(SC13) MB8364-005
M164(SC14) MB8364-006
M165(SC15) MB8364-007
M166(SC16) MB8364-008
M167(SC17) MB8364-009
M168(SC18) MB8364-010
M169(SC19) MB8364-011
KA0 KA1 KA2 KA3 KA4 KA5 KA6 KA7 KA8 KA9 KA10 KA11 KA12
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 CE
O1 O2 O3 O4 O5 O6 O7 O8
KDH0 KDH1 KDH2 KDH3 KDH4 KDH5 KDH6 KDH7
KDL0 KDL1 KDL2 KDL3 KDL4 KDL5 KDL6 KDL7

漢字ROM(3)

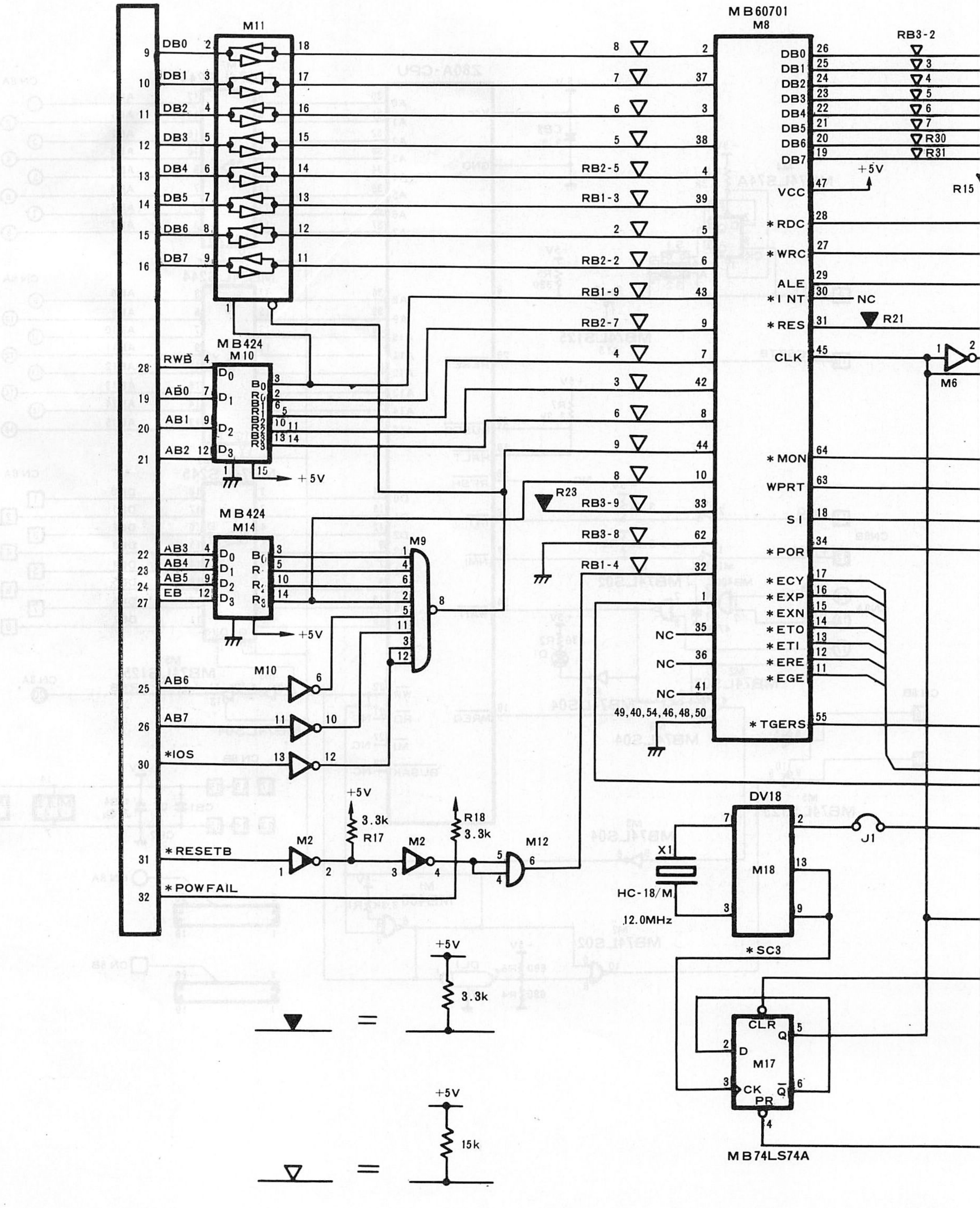


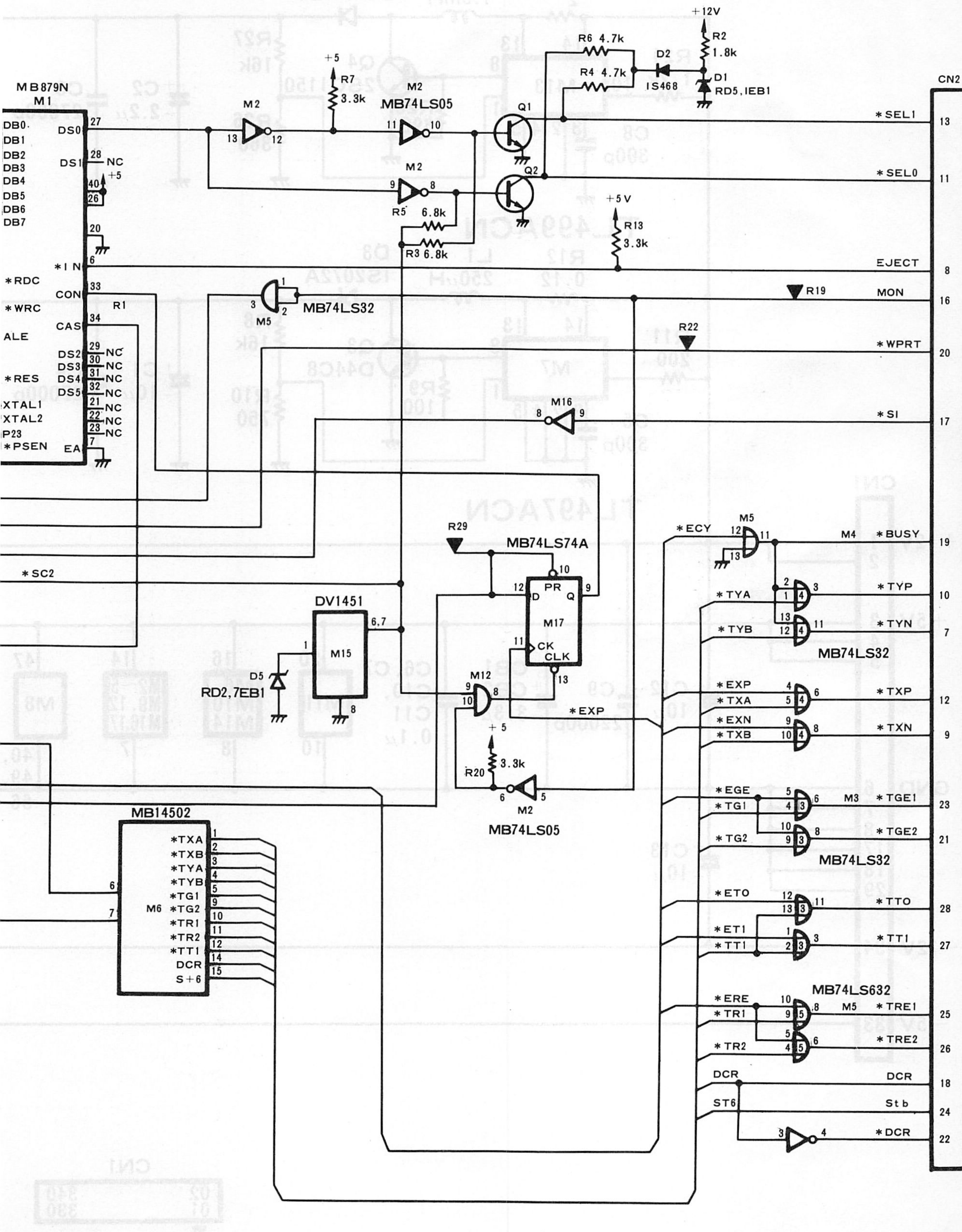
M170(SC20)
MB8364-012
M171(SC21)
MB8364-013
M172(SC22)
MB8364-014
M173(SC23)
MB8364-015
M174(SC24)
MB8364-016
M175(SC25)
MB8364-017
M176(SC26)
MB8364-018
M177(SC27)
MB8364-019
(20-A9) KADDRESS
KDATAH (20-D7)
KDATAL (20-F7)
(20-C9) KCE 4
(20-C9) KCE 5
(20-C9) KCE 6
(20-C9) KCE 7
KA0 KA1 KA2 KA3 KA4 KA5 KA6 KA7 KA8 KA9 KA10 KA11 KA12
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 CE
O1 O2 O3 O4 O5 O6 O7 O8
KDH0 KDH1 KDH2 KDH3 KDH4 KDH5 KDH6 KDH7
KDL0 KDL1 KDL2 KDL3 KDL4 KDL5 KDL6 KDL7

Z80カード

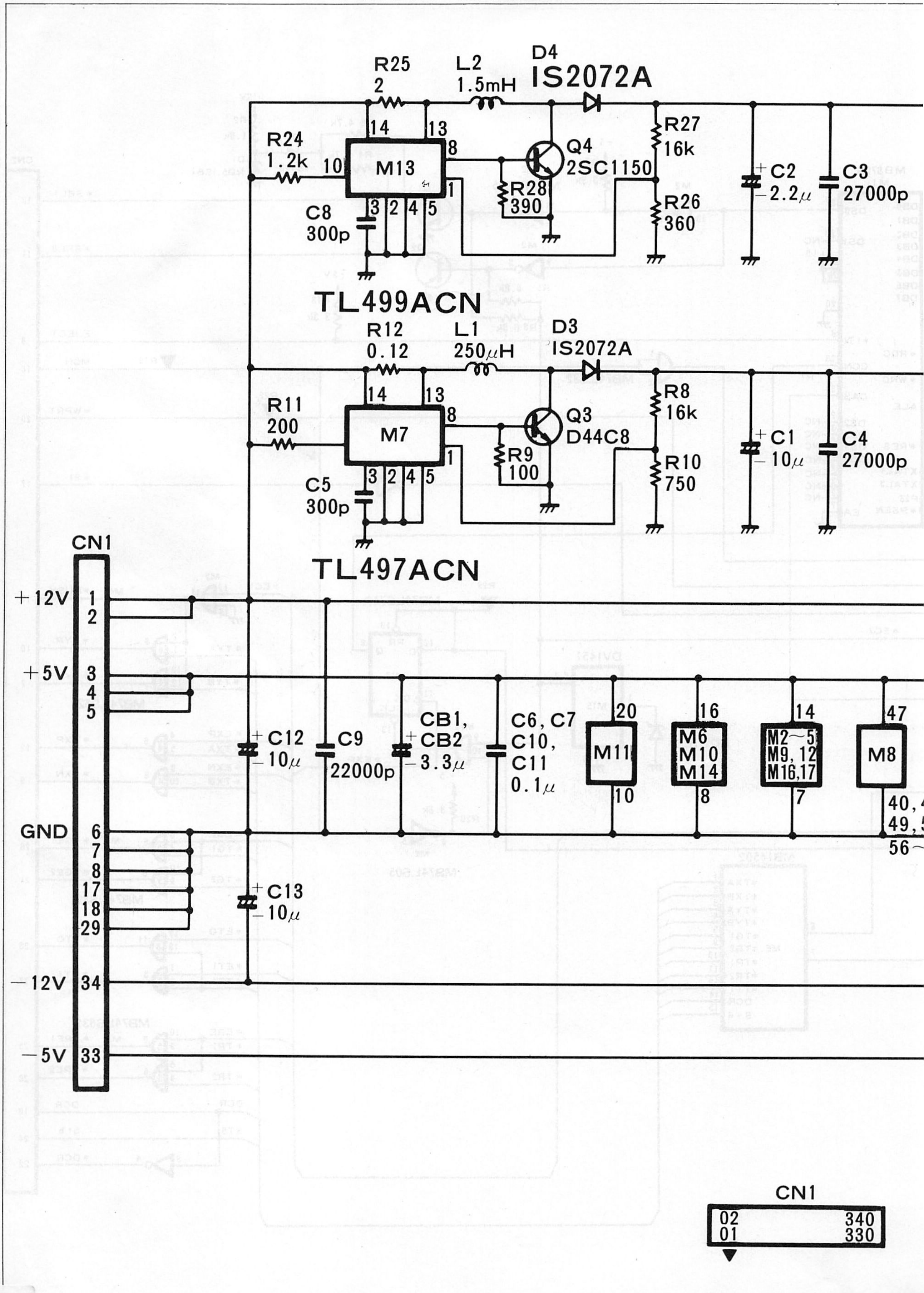

Z80A-CPU
MB74LS74A
MB74LS125
MB74LS244
MB74LS245
MB74LS02
MB74LS04
MB400
CN 8A
CN 8B
*RESETB
*IRQ
*NMI
BAB
BSB
Z80φ
RWB
AB0
AB15
DB0
DB7

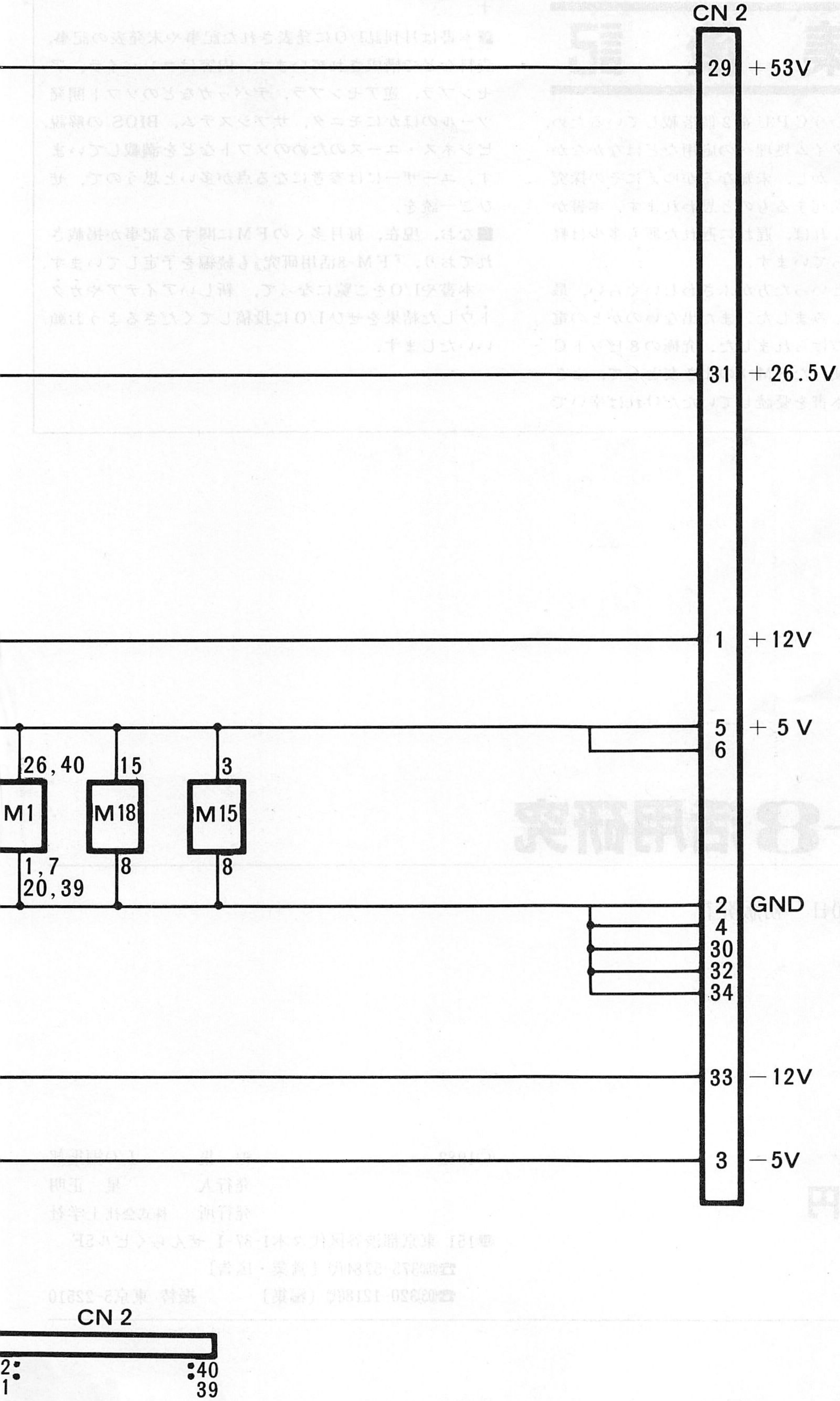
バブルメモリ・コントロールカード(1)





MB879N
M1
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
DS0
DS1
NC
+5
*IN
*RDC
CON
*WRC
R1
CAS
ALE
DS2
DS3
DS4
DS5
*RES
XTAL1
XTAL2
P23
*PSEN
EA
M2
MB74LS05
R7
3.3k
Q1
Q2
R6 4.7k
R4 4.7k
D2
IS468
+12V
R2
1.8k
D1
RD5,IEB1
R5
6.8k
R3 6.8k
+5V
R13
3.3k
M5
MB74LS32
R19
R22
M16
*SC2
R29
MB74LS74A
PR
D
Q
M17
CK
CLK
DV1451
M15
D5
RD2,7EB1
M12
*EXP
R20
3.3k
MB74LS05
MB14502
*TXA
*TXB
*TYA
*TYB
*TG1
M6
*TG2
*TR1
*TR2
*TT1
DCR
S+6
*ECY
*TYA
*TYB
*EXP
*TXA
*EXN
*TXB
*EGE
*TG1
*TG2
*ETO
*ET1
*TT1
*ERE
*TR1
*TR2
DCR
ST6
M4
M3
MB74LS632
CN2
*SEL1
*SEL0
EJECT
MON
*WPRT
*SI
*BUSY
*TYP
*TYN
*TXP
*TXN
*TGE1
*TGE2
*TTO
*TT1
*TRE1
*TRE2
DCR
Stb
*DCR


R25
2
L2
1.5mH
D4
IS2072A
R24
1.2k
M13
Q4
2SC1150
R27
16k
R28
390
R26
360
C8
300p
C2
2.2μ
C3
27000p
TL499ACN
R12
0.12
L1
250μH
D3
IS2072A
R11
200
M7
Q3
D44C8
R8
16k
R9
100
R10
750
C5
300p
C1
10μ
C4
27000p
TL497ACN
CN1
+12V
+5V
GND
−12V
−5V
C12
10μ
C9
22000p
CB1, CB2
3.3μ
C6, C7
C10,
C11
0.1μ
M11
M6
M10
M14
M2〜5
M9, 12
M16,17
M8
C13
10μ
CN1
02
01
340
330



CN 2
29 +53V
31 +26.5V
1 +12V
5 +5V
6
26,40
15
3
M1
M18
M15
1,7
20,39
8
8
2 GND
4
30
32
34
33 −12V
3 −5V
CN 2
2
1
40
39

編集後記

■FM-8は6809というCPUを2個搭載しているため，その解析やリアルタイム処理への応用などはなかなか困難なものです．しかし，未知なるがゆえにその探究は一種の好奇心を喚起するものと思われます．本書がそのよき手引書となれば，遅れに遅れた罪も多少は軽くなるのではと思っています．

■FM-8格闘研究といった方がふさわしいぐらい，最後の最後まで，苦しみました．まだ出ないのかとの電話にたいへん元気づけられました．究極の8ビットCPU6809を2個も使ったFM-8のよき友として，よきバイブルとして，本書を愛読していただければ幸いです．

■本書は月刊誌I/Oに発表された記事や未発表の記事，資料などで構成されています．内容はコンパイラ，アセンブラ，逆アセンブラ，デバッガなどのソフト開発ツールのほかにモニタ，サブシステム，BIOSの解説，ビジネス・ユースのためのソフトなどを満載しています．ユーザーには参考になる点が多いと思うので，ぜひご一読を．

■なお，現在，毎月多くのFMに関する記事が掲載されており，『FM-8活用研究』も続編を予定しています．

本書やI/Oをご覧になって，新しいアイデアやカクトウした結果をぜひI/Oに投稿してくださるようお願いいたします．

I/O別冊

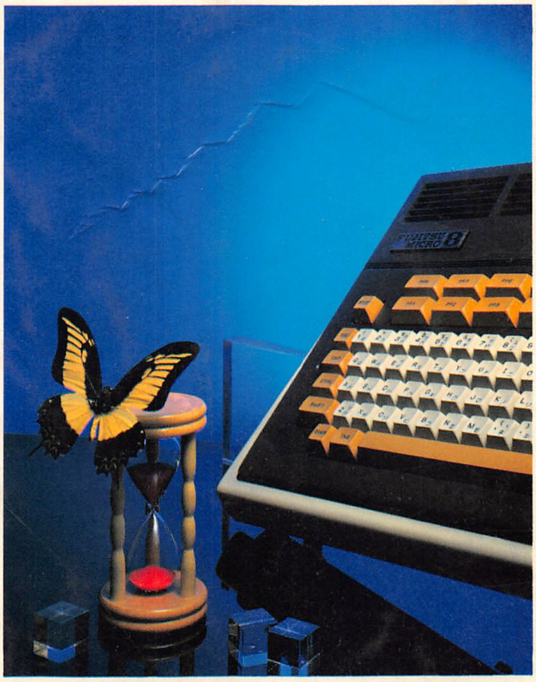
FM-8活用研究

昭和57年7月20日　初版発行

定価2500円

編　集　I/O編集部
発行人　星　正明
発行所　株式会社工学社
〒151 東京都渋谷区代々木1-37-1 ぜんらくビル5F
☎(03)375-5784(代)〔営業・広告〕
☎(03)320-1218(代)〔編集〕　振替 東京5-22510

印刷：㈱耕文社



FM-8活用研究

定価2500円

雑誌61473-46